A Hybrid Feature Selection Framework Using Opposition-Based Harmony Search and Manta Ray Foraging Optimization

Thatikonda Somashekar [1,*] and Srinivas Jagirdar [2]

[1] Department of Computer Science and Engineering, University College of Engineering, Osmania University, Hyderabad, India
[2] Department of Information Technology, Matrusri Engineering College, Hyderabad, India
Email: soma.ts@gmail.com (T.S.); drjsrinivas@matrusri.edu.in (S.J.)
*Corresponding author

***Abstract*—Feature selection is the process of extracting an optimal subset feature from a primary feature set to minimize data dimensionality. The hybrid metaheuristic is the most common method for dealing with optimization problems. This manuscript proposes a hybrid of Opposition-based Harmony Search (OBHS) and Manta Ray Foraging Optimization (MRFO) for feature selection, which is one of the human-based metaheuristic optimization algorithms. The proposed OBHS-MRFO methodology's experiments are tested on 21 benchmark datasets taken from the University of California, Irvine (UCI) repository. This dataset is split into three classes: low, medium, and high-scale based, on the dataset dimensions. The proposed model is utilized to overcome the issues of minimum accuracy produced by redundant and irreverent features. The obtained result is compared to four algorithms namely, FS-BGSK, FS-pBGSK, OBHS, and MRFO algorithms. It concludes that the proposed OBHS-MRFO algorithm obtains better results when compared with other methods with regard to average fitness function value, average accuracy, average feature selection size, standard deviation, and computational time.**

***Keywords*—feature selection, machine learning, manta ray foraging optimization, metaheuristic algorithm, opposition-based harmony search**

I. INTRODUCTION

Feature selection is an important technique in Machine Learning (ML) and data mining, utilized to select relevant and important features from high-dimensional data. It has various real-world applications in several domains like medicine, document modeling, biology, representational and multi-view learning [1]. The purpose of selecting features is to determine an optimal feature subset from the initial dataset that enhances the performance of ML by minimizing dimensionality, eliminating irrelevant features, and saving resources [2]. The feature selection method contains three classes namely, filter, embedded, and wrapper method [3]. The filter method has a low computational rate and avoids overfitting issues that arise from ignoring upcoming dependencies, restricted insensitivity, and adaptability to classification boundaries [4]. The embedded method robotically selects features for the optimal subset of the model, saving time and effort. Incorporating a model for the feature selection process efficiently decreases the overfitting issues, but also maximizes the computational power [5]. The wrapper method captures non-linear features and interaction relationships between various features by adapting feature selection for a particular ML algorithm which has high-dimensional data and is computationally expensive [6].

The robustness of the feature selection technique is evident in two of its procedures, i.e., search and evaluation. Selecting the respective features from the original set with every subset results in a combinatory explosion [7]. Furthermore, search methods are known to select valuable features effectively where they utilize backward and forward traditional greedy search strategies [8]. The issues with this category of searching are local optima which result in non-optimal features. The estimation function handles these problems by evaluating every feature subset, thereby helping to determine the global optimal solutions [9]. There is a feature selection issue established as NP-complete combinatory optimization issue, wherein the optimal features are chosen from the initial data [10]. The two objectives by name of minimizing dimensionality and maximizing performance are considered multi-objective problems to minimize the dimension from the initial data and gain the highest performance with a low quantity of features [11]. To resolve these types of issues, a metaheuristic algorithm is preferable for better performance [12]. The main challenge in feature selection is balancing the tradeoff between accuracy and cost. Utilizing a metaheuristic algorithm efficiently balances this conflict by leveraging metaheuristic data to narrow down the search space [13]. The hybridization of Opposition-Based Harmony Search (OBHS) and Manta Ray Foraging Optimization (MRFO) for feature selection aims to provide a significant contribution by combining the unique strengths of each algorithm. The integration of

Manuscript received December 2, 2023; revised January 25, 2024; accepted February 21, 2024; published August 28, 2024.

doi: 10.12720/jait.15.8.982-990

OBHS with its opposition-based mechanism promotes enhanced exploration, while the MRFO leverages manta ray foraging behavior for efficient exploitation, and creates a balanced approach that improves the overall feature selection process. The hybrid algorithm yields a much better exploration-exploitation balance, resulting in solutions of higher quality and adaptability to diverse problem characteristics. By overcoming individual limitations and harnessing the synergy between OBHS and MRFO, the hybridization offers the potential to generate diverse and high-quality feature subsets, making it a promising solution for addressing various challenges in feature selection tasks. The main contributions of the manuscript are as follows:

- Hybridization combines OBHS's opposition-based exploration with MRFO's efficient exploitation for a balanced feature space search.
- Improved feature subset quality and adaptability, as well as diverse dataset characteristics are attained through the integration of OBHS and MRFO.
- Aims to overcome the individual algorithms' limitations, by utilizing OBHS's diversity and MRFO's exploitation for solution generation in feature selection tasks.

The remaining parts of the paper are organized as follows: Section II provides the literature review. Section III provides details of the proposed methodology. Section IV details the results and discussion, while Section V gives the conclusion.

II. Literature Review

Pashaei and Pashaei [14] introduced an efficient binary approach for feature selection, using two Chimp Optimization Algorithms (ChOA), in medical data classification. In the first algorithm, dual transfer functions (S and V-shaped) were utilized to alter continuous data into binary form. In the next algorithm, a crossover operator was utilized to improve the exploratory behavior of ChOA. To validate the proposed method's efficiency, five publicly available datasets were employed from various domains like text, image, and life. The developed model was utilized to decide the possible solution for difficult optimization problems due to its high convergence rate, and simplicity. The developed model was not an iterative method and its search strategy contained one iteration only, which easily got stuck in the local optima.

Sun *et al.* [15] developed a hybrid feature selection approach using improved Sine Cosine Algorithm with metaheuristic techniques to minimize the data dimensionality due to a huge amount of features in the dataset. When analyzing the performance of optimization, it was seen that the standard SCA algorithm had complexity during feature selection in selecting the better feature subset. The performance of the developed algorithm was evaluated on the UCI dataset and compared with three algorithms, SCA, PSO, and WOA. The developed method had advantages of low computational complexity, alongside the resolving of feature selection in huge-dimension datasets. However, the limitations such as propensity to fall into the local optimum and slow convergence rate were noted.

Wu *et al.* [16] implemented a threshold Binary Grey Wolf Optimizer-based Multi-elite interaction for feature selection (MTBGWO). The numerous subset topology for global search was employed, where the elite wolf learned to acquire another elite position by interchanging further information among subsets and taking multiplicity into their subsets. At last, the threshold technique was utilized to alter the continual position of individual grey wolves into binary for application in feature selection problems. The MTBGWO technique utilized multi-elite information interaction method which enhanced the local exploitation ability and decreased the convergence speed.

Xu *et al.* [17] presented a multiple Binary Arithmetic Optimization Algorithm (BAOA) which employed various strategies for performing feature selection. Initially, six algorithms were formed by altering continuous search space into random search space based on six transfer functions. Additionally, six more algorithms were established by incorporating Levy flight and transfer functions to enhance the searching speed and escaping ability from local optima. The developed model's performance was evaluated by employing various strategies on the UCI dataset. The model had advantages like easy implementation for better feature selection. But, this model had weak search proficiency and a slow convergence rate.

Wang *et al.* [18] suggested a Crisscross Harris Hawks Optimizer (CCHHO) by innovatively using the horizontal and vertical crisscross strategy of Criss-cross Optimization Algorithm (CSO) in HHO for controlling the global tasks and feature selection problems. In CCHHO, the CSO vertical and horizontal crossover strategy was utilized for altering the exploitative capability to improve local optimum. The CSO horizontal crossover approach was taken as an operator for enhancing explorative tendency. It enhanced the balance between exploration and exploitation abilities, and an aggressive operator was assumed to simulate the convergence rate. The advantage of using this model was the improvement of problem dimensions. Nonetheless, this model spent more time on execution.

Sun *et al.* [19] introduced a rolling bearing fault feature selection technique-based clustering hybrid binary cuckoo search approach for extracting the time-frequency. A clustering hybrid initialization was utilized in a population to minimize the redundant features. This method utilized a Louvain algorithm to cluster features and initialize the population based on the number of features and clustering information. The Levy flight-based mutation strategy efficiently utilized high-quality population information by managing various high-quality individuals. The developed method efficiently minimized the redundant features by utilizing clustering data and a hybrid initialization approach. When resolving feature selection problems, this model exhibited a slow convergence rate and random initialization.

Zaimoglu *et al.* [20] developed a Binary Chaotic Horse Herd Optimization Algorithm for Feature Selection (BCHOAFS) before adding chaotic maps. The introduced model was utilized to select the combination of optimal features that reduced accuracy in classification by reducing the number of selected features. The SMF operator was established as a local search approach to enhance both exploration and exploitation abilities, whereas the ML classification algorithm was utilized to test the reduced subset accuracy. The introduced model had advantages like eliminating redundant, irrelevant, noisy data and enhancing the performance of the learning algorithm. Nonetheless, this introduced model had low accuracy and low convergence rate.

Fang and Liang [21] implemented a hybrid technique based on a Nonlinear Binary Grasshopper Whale Optimization Algorithm (NL-BGWOA) to resolve feature selection problems. The introduced method incorporated the whale individuals updating technique in WOA with GOA, as well as optimized the position updating approach which optimized the searching diversity in the target domain. The NL-BGWOA integrated nonlinear adjustment coefficients and adaptive weights. It also minimized the computational cost and size of the data space. The core coefficient was directly reduced at a persistent value which in turn minimized the convergence rate and fell into the local optimum easily.

Agrawal *et al.* [22] suggested a feature selection technique through S and V shaped Gaining-Sharing Knowledge (GSK) algorithm. The population reduction system was utilized through the transfer function to increase the model's performance. Hence, the search space was discovered and the worst solution from the search space was removed because of the population size in each iteration. The suggested approach was utilized to remove the non-feasible solution at the primary stage without prompting the exploration ability. Yet, this model failed to designate a suitable population size.

Agrawal *et al.* [23] presented Feature Selection technique through Binary GSK (FS-BGSK) algorithm. The FS-population reduction on BGSK (FS-pBGSK) utilized BGSK to increase the exploration and exploitation quality of FS-BGSK. The presented approach utilized the 22-feature selection benchmark dataset from UCI repository. The FS-BGSK was deployed to select an appropriate population size, and protect against premature convergence and local optimum issues. Anyhow, this model attained lesser accuracy and higher computational time.

III. Proposed Method

This proposed method is utilized for feature selection on the University of California, Irvine (UCI) dataset which contains low, medium, and high dimensional datasets. The preprocessing using min-max normalization enhances the model's performance and it is given as input to the feature selection process. The hybrid OBHS and MRFO are utilized for feature selection. Then, the selected features are given as input to the classification process which includes K-Nearest Neighbour (KNN), Extreme Learning Machine (ELM), Multi-layer Perceptron (MLP), and Support Vector Machine (SVM). The performance is evaluated based on the average fitness function value, average accuracy, average feature selection size, standard deviation, and computational time. The block diagram of the proposed methodology is represented in Fig. 1.

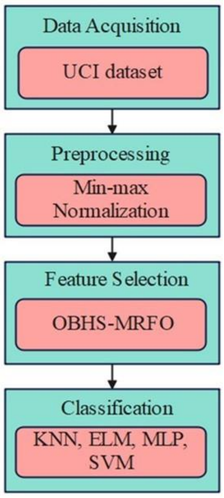


Fig. 1. Block diagram of the proposed methodology.

A. Data Acquisition

In this manuscript, the proposed methodology's experiments were tested on the 21-benchmark dataset taken from University of California, Irvine (UCI) repository [24]. This dataset is split into three classes named low, medium, and high-scale based on the dataset dimensions. The dataset includes several numbers of features ranging from 9–856, and numerous instances ranging from 32–5000 in the feature selection technique. The equivalent measure of every dataset is considered for training, testing, and validation sets in the rotation estimation aspect. The presentation and description of each dataset are presented in Table I.

TABLE I. The Description of the UCI Benchmark Dataset

Type	Dataset	Features	Instances
Low Dimension L1–L8	Tic-Tac-Toe	9	958
	Breast Cancer	10	699
	Wine	13	179
	Heart	13	270
	Zoo	16	101
	House-vote	16	435
	Lymphography	18	148
	Hepatitis	19	155
Medium Dimension M1–M8	Waveform	21	5000
	German	24	1000
	Wdbc	30	569
	Soybean	34	47
	Dermatology	34	366
	Ionosphere	34	351
	Lung cancer	56	32
	Sonar	60	208
High Dimension H1–H5	Hillvalley	100	606
	Clean	165	476
	Semeion	265	1593
	Arrhythmia	279	452
	CNAE	856	1080

B. Preprocessing

The University of California, Irvine (UCI) repository dataset is standardized using the min-max normalization method [25]. Here, the maximum score of feature is changed into 1, the minimum score of feature is changed into 0 and the remaining feature is changed into an integer among 0 and 1, which is presented in Eq. (1):

$$f(x,y)=\frac{f(x,y)-Z_{min}}{Z_{max}-Z_{min}} \tag{1}$$

where, f is the input image, x and y are pixel locations in the image, Z_{max} and Z_{min} are correspondingly the maximum and minimum pixel values. After preprocessing, the features are selected through an optimization algorithm.

C. Feature Selection

OBHS introduces an opposition-based mechanism to enhance exploration by pairing solutions with their opposites. Meanwhile, MRFO is inspired by the foraging behavior of manta rays, aiming for a balanced exploration-exploitation strategy. Hybridizing OBHS and MRFO for feature selection seeks to leverage the strengths of both algorithms. The opposition-based approach of OBHS enhances exploration while MRFO's foraging behavior contributes to efficient exploitation. Nevertheless, as individual algorithms, both OBHS and MRFO have limitations. OBHS suffers from premature or slow convergence, and its effectiveness depends on problem characteristics and parameter settings. MRFO on the other hand, requires careful parameter tuning and does not perform optimally in certain situations. The MRFO is a bio-inspired optimizer that simulates the food searching behavior of manta rays. The manta ray is a familiar sea organism that feeds on plankton and small animals in water. The manta rays contain three food-finding procedures called chain, cyclone, and somersault foraging [26]. These three foraging processes form a primary search scheme of the MRFO.

1) Chain foraging

The chain foraging imitates the process of essential food search behavior where the manta rays analytically grasp up to catch undetected or disappeared prey in the chain by the previous manta rays. The cooperative collaboration between the challenging manta rays removes the probable prey loss in their eyesight and enhances the food rewards. The MRFO assumes that a better solution is attained through the plankton with maximum absorption of a target plankton for the manta ray chain. This process updates the present position of individual populations based on the target prey, giving a better result. The chain foraging update mechanism is shown in Eqs. (2) and (3):

$$X_{i,j}^{t+1}=\begin{cases} X_{i,j}^{t}+r_1(0,1)\left(X_{best,j}^{t}-X_{i,j}^{t}\right)+\alpha\left(X_{best,j}^{t}-X_{i,j}^{t}\right) i=1 \\ X_{i,j}^{t}+r_2(0,1)\left(X_{i-1,j}^{t}-X_{i,j}^{t}\right)+\alpha\left(X_{best,j}^{t}-X_{i,j}^{t}\right) i=2,\ldots,N \end{cases} \tag{2}$$

$$\alpha=2\times r_3(0,1)\times\sqrt{\left|\log\left(r_4(0,1)\right)\right|} \tag{3}$$

where $X_{i,j}^{t}$ is i^{th} position of manta ray in dimension j, iteration t. $r_i(0,1) i=1,2,3,4$ is a random number ranging between $[0,1]$ that is dissimilar from one another. α is the coefficient weight and X_{best}^{t} is the highest attention of prey. In chain foraging, the position update mechanism is established through the last manta ray in the chain and spatial location of the prey.

2) Cyclone foraging

Cyclone foraging occurs when the collection of amount of the plankton is important. The manta ray's head is paired with its tail, making a spiral to generate an edge in the cyclone's eye, thus the filtered water transfers to the surface. This difficult procedure allows the prey, plankton to simply grab the manta ray predator. A group of manta rays generate a foraging chain and produce spatial movements as the food source approaches when identifying the school plankton position in deep water. In cyclone foraging, the grouped manta rays not only track the manta ray before the last one so as to confirm the developed chain constancy, but chase a pathway of spiral change towards the target plankton. The manta ray movement in chain spiral shape is statistically displayed in D dimensional search space as given in Eqs. (4) and (5):

$$X_{i,j}^{t}=\begin{cases} X_{best,j}^{t}+r_5(0,1)\left(X_{best,j}^{t}-X_{i,j}^{t}\right)+\beta\left(X_{best,j}^{t}-X_{i,j}^{t}\right) & i=1 \\ X_{best,j}^{t}+r_6(0,1)\left(X_{i-1,j}^{t}-X_{i,j}^{t}\right)+\beta\left(X_{best,j}^{t}-X_{i,j}^{t}\right) & i=2,\ldots,N \end{cases} \tag{4}$$

$$\beta=2e^{r_7(0,1)\left(\frac{maxitr-itr+1}{maxitr}\right)}\times\sin\left(2\pi r_8(0,1)\right) \tag{5}$$

where β is the coefficient weight, $maxitr$ and itr represent the maximum iteration and present iteration, respectively. $r_i(0,1) i=5,6,7,8$ are different random numbers in range of $[0,1]$. The cyclone process is a significant part of executing two mechanisms, exploitation and exploration. Employing better prey in the foraging phase overlays the way of increasing generative areas from the present best solution, and provides the exploitation ability of the algorithm.

Moreover, the cyclone foraging generates an important contribution to the exploration stage by applying individual populations to transfer a random position in the Harmony Memory (HM) which generates the best prey position. The primary HM has Harmony Memory Size (HMS) number of n-dimension that randomly provides probable solutions, where the representation of HMS is formulated in Eq. (6):

$$HM=\left[X^1X^2X^3\ldots X^{HMS}\right]'\in R^{HMS\times n} \tag{6}$$

where, $X^p=\left[x_1x_2x_3\ldots x_p\right]$ and $p\in[1,n]$.

3) Somersault foraging

Somersault foraging is the final foraging process and one of the most splendid, yet usual activities created by humanity. Manta rays perform a circular movement and backward rotation around the plankton prey, thereby moving them into their open mouths. This process decides the best prey position in the axis and every manta ray in the population analyzes the point to a new location in the search field. The solution update approach is inclined through the present best solution that guides the designed individual chain through Eq. (7):

$$X_{i,j}^{t+1} = X_{i,j}^{t} + sf \times (r_9(0,1) \times X_{best,j}^{t} - r_{10}(0,1) \times X_{i,j}^{t} \tag{7}$$

where, sf is the somersault factor equivalent to 2 and r_9, r_{10} are random numbers within the range of [0,1]. The MRFO algorithm is initialized by producing an individual random population with quantified restrictions. The position update depends on the manta ray individual in front of the current one and the considered pivot point. Altering from exploration to exploitation phase relies on the differences in the arithmetic value of (*itr/maxitr*) ratio. The exploitation is defined when $itr/maxitr < r(0,1)$. In that, the present best location is measured on an axis that plays the main part in candidate solutions near the promising and fertile regions in a search domain. Moreover, it changes among the chain and cyclone foraging acquired on randomly formed numbers. Then, this foraging is considered to update the individual present position. These three foraging mechanisms are established to achieve the global optimum solution of optimization problems and satisfy the predetermined end condition.

D. Classification

After feature selection, the selected features are classified by using different classification techniques namely, KNN, ELM, MLP, and SVM.

1) KNN

The KNN classifier is an instance-based, non-parametric model which is employed for classification [27]. The KNN is a supervised learning algorithm which categorizes the unidentified samples through evaluating the distance among certain unidentified samples and the nearest k-neighbor. The k-nearest samples are selected once all the distances are evaluated, wherein the majority class among KNN provides a prediction for the latest record. The KNN classifier is a frequently used method for classifying features as it is simple to implement. In general, the Euclidean distance is utilized to evaluate the distance which is formulated in Eq. (8):

$$|X_1 - X_2| = \left[\sum_{i=1}^{d} (x_{1i} - x_{2i})^2 \right]^{1/2} \tag{8}$$

where $|X_1 - X_2|$ is the distance between X_1 and X_2 which are the dimension points in d, x_{1i} and x_{2i} are the horizontal and vertical axes values, respectively in the coordinate plane.

2) ELM

The ELM is an ML technique utilized for supervised learning tasks in terms of classification. The ELM is a type of feed-forward neural network that stands out for its fast training speed and simplicity [28]. Unlike traditional neural networks, both the input and hidden layer weights are iteratively adjusted during training. The ELM fixes the input layer weights and learns the output layer weights. It differs from the MLP, wherein the ELM initializes weights and biases randomly, and the last output is attained by linear combination. It is widely employed due to its effective and fast learning speed, fast convergence, better ability for generalization, and ease of execution.

3) MLP

MLP is a feedforward neural network [29] method that maps the input data to a set of suitable outputs. The MLP contains numerous layers where all layers are interconnected with each other by weights. Except for the input layer nodes, all other layer nodes are represented as neurons with nonlinear activation functions. There is more than one nonlinear hidden layer between the input and output layers. The interrelated neurons in MLP are in one-directional fashion, and the weights of connection are within the range of [−1, 1].

4) SVM

The SVM is one of the supervised ML techniques, having larger classification efficiency in comparison to various other classification models [30]. However, the implementation of SVM is limited because of the need for high training time for larger data. The SVM, when integrated with feature selection techniques, acquires reduced dimension data. Moreover, the SVM method has the advantage of effective analysis of the non-linear relationship between the processes and features of the data, so as to provide an efficient classification. The benefit of utilizing SVM is its memory effectiveness and efficiency in large dimensional space which is applied for regression. The SVM separates data by finding the better hyperplane and margin. The hyperplane is a plane differentiator among two classes, whereas the margin is the distance between the outermost instances of the class.

IV. RESULT

In this paper, the proposed OBHS-MRFO is simulated on Python environment with a system configuration of 16 GB RAM, Windows 10 operating system, and intel core i5. The parameters of average fitness function, accuracy, feature selection size, standard deviation, and computational time are employed to estimate the model's performance. The mathematical formulae of these parameters are presented in Eqs. (9)–(13), respectively.

- **Average fitness function value**

$$Avg_Z = \frac{1}{W} \sum_{i=1}^{W} z_i^* \tag{9}$$

where, Avg_Z is the fitness function mean over W, and z_i^* is the optimum fitness score on i^{th} run.

- **Average accuracy**

$$Avg_{Acc} = \frac{1}{W} \sum_{i=1}^{W} Acc_i^* \tag{10}$$

where, Avg_{Acc} is the average classification accuracy over W, and Acc_i^* is the optimal accuracy on the i^{th} run.

- **Average feature selection size**

$$Avg_{feature} = \frac{1}{W}\sum_{i=1}^{W}\frac{length(x)_i^*}{|D|} \quad (11)$$

where $Avg_{feature}$ is the average selection size of features over W, $length(x)_i^*$ is the determined feature length on the i^{th} run, and $|D|$ is the feature counts.

- **Standard deviation of fitness value**

$$Std_Z = \left[\frac{1}{W-1}\sum_{i=1}^{W}(z_i^* - Avg_Z)^2\right]^{\frac{1}{2}} \quad (12)$$

where Std_Z is the standard deviation, z_i^* is optimal fitness score on the i^{th} run, and Avg_Z is the fitness function mean.

- **Average computational time**

$$Avg_{Time}^o = \frac{1}{W}\sum_{i=1}^{W}(Time)_i^o \quad (13)$$

where Avg_{Time}^o is the average computational time, and $(Time)_i^o$ is the time utilized by the i^{th} run in o^{th} algorithm.

A. *Quantitative Analysis*

The quantitative analysis of OBHS-MRFO on the feature selection dataset is evaluated by utilizing state-of-art algorithms which are, FS-BGSK, FS-pBGSK, OBHS, and MRFO. The performances of these algorithms are measured and matched with that of the OBHS-MRFO algorithm. The population size diminishes linearly through the number of function evaluations. Both binary stages allow the OBHO-MRSO to discover search space and find optimal solutions.

TABLE II. AVERAGE FITNESS FUNCTION VALUES FROM VARIOUS OPTIMIZERS

Datasets	FS-BGSK	FS-pBGSK	OBHS	MRFO	OBHS-MRFO
L1	0.2097	0.1974	0.19216	0.18489	0.17608
L2	0.0299	0.0322	0.02760	0.02284	0.01770
L3	0.1498	0.1455	0.14043	0.13767	0.13034
L4	0.0409	0.0404	0.03472	0.03240	0.02883
L5	0.0438	0.0440	0.03903	0.03791	0.03560
L6	0.0521	0.0423	0.03865	0.02942	0.02807
L7	0.5037	0.4894	0.48239	0.47827	0.47324
L8	0.2690	0.2400	0.23030	0.22577	0.21773
M1	0.1701	0.1684	0.16258	0.15620	0.15587
M2	0.2462	0.2372	0.22951	0.21968	0.21938
M3	0.0515	0.0480	0.04419	0.04018	0.03431
M4	0.0945	0.0762	0.07484	0.07001	0.06498
M5	0.0108	0.0074	0.00677	0.00676	0.00672
M6	0.0009	0.0006	0.00050	0.00044	0.00039
M7	0.1128	0.0813	0.08048	0.07381	0.06996
M8	0.0754	0.0698	0.06757	0.06438	0.05848
H1	0.4071	0.3861	0.38147	0.37885	0.39952
H2	0.0565	0.0530	0.05250	0.04658	0.04264
H3	0.0134	0.0134	0.00726	0.00722	0.00124
H4	0.2871	0.2820	0.28053	0.27900	0.27558
H5	0.0057	0.0046	0.00457	0.00454	0.00453

Table II represents the average fitness function values from various optimizers evaluated on the datasets of L1–L8, M1–M8, and H1–H5. In that, OBHS-MRFO attains better results on H1–H5 dataset with the values of 0.39952, 0.04264, 0.00124, 0.27558, and 0.00453, correspondingly.

Table III represents the average accuracy achieved by different algorithms evaluated on the datasets of L1–L8, M1–M8, and H1–H5. From Table III, it is evident that OBHS-MRFO attains superior results on the H1–H5 dataset with values of about 0.62748, 0.96335, 1.00000, 0.73343, and 1.00000, respectively.

TABLE III. AVERAGE ACCURACY ACHIEVED BY DIFFERENT ALGORITHMS

Datasets	FS-BGSK	FS-pBGSK	OBHS	MRFO	OBHS-MRFO
L1	0.7989	0.8082	0.81065	0.82060	0.82563
L2	0.9751	0.9726	0.97509	0.97950	0.98137
L3	0.8524	0.8566	0.86355	0.86814	0.87180
L4	0.9622	0.9627	0.96743	0.96874	0.97527
L5	0.9567	0.9563	0.95801	0.96249	0.96809
L6	0.9513	0.9616	0.97116	0.97982	0.98343
L7	0.4952	0.5094	0.51330	0.51494	0.52030
L8	0.7313	0.7610	0.76910	0.77360	0.78172
M1	0.8358	0.8373	0.84448	0.84458	0.84604
M2	0.7557	0.7646	0.77153	0.77414	0.78145
M3	0.9495	0.9526	0.95653	0.95757	0.96296
M4	0.9074	0.9250	0.92867	0.93695	0.94375
M5	0.9934	0.9962	1.00540	1.00000	1.00000
M6	1.0000	1.0000	1.00000	1.00000	1.00000
M7	0.8875	0.9188	0.91986	0.92094	0.92361
M8	0.9284	0.9338	0.93963	0.94642	0.95384
H1	0.5927	0.6139	0.61759	0.62539	0.62748
H2	0.9475	0.9508	0.95248	0.95981	0.96335
H3	0.9905	0.9905	0.99662	1.00000	1.00000
H4	0.7137	0.7190	0.72421	0.72501	0.73343
H5	0.9961	0.9972	1.00000	1.00000	1.00000

TABLE IV. AVERAGE SELECTED FEATURE RATIO TO NUMBER OF TOTAL FEATURES

Datasets	FS-BGSK	FS-pBGSK	OBHS	MRFO	OBHS-MRFO
L1	0.72	0.75	0.74944	0.74507	0.73930
L2	0.52	0.50	0.49711	0.48939	0.48768
L3	0.37	0.35	0.34493	0.34021	0.33611
L4	0.35	0.34	0.33466	0.32671	0.31899
L5	0.10	0.08	0.07418	0.07076	0.06809
L6	0.39	0.42	0.41137	0.41026	0.40301
L7	0.40	0.38	0.37581	0.37411	0.36494
L8	0.30	0.34	0.33083	0.33015	0.32365
M1	0.76	0.73	0.72405	0.71789	0.71182
M2	0.43	0.42	0.41334	0.41269	0.40956
M3	0.14	0.11	0.10810	0.09879	0.09017
M4	0.28	0.19	0.18066	0.17342	0.16522
M5	0.43	0.36	0.35311	0.34556	0.33844
M6	0.09	0.06	0.05158	0.04700	0.04020
M7	0.14	0.09	0.08464	0.07732	0.06835
M8	0.45	0.43	0.42686	0.42389	0.41469
H1	0.39	0.38	0.37407	0.36511	0.39830
H2	0.45	0.43	0.42570	0.42268	0.41336
H3	0.39	0.39	0.38429	0.37774	0.36912
H4	0.36	0.38	0.37554	0.37484	0.36583
H5	0.18	0.18	0.17516	0.16575	0.15783

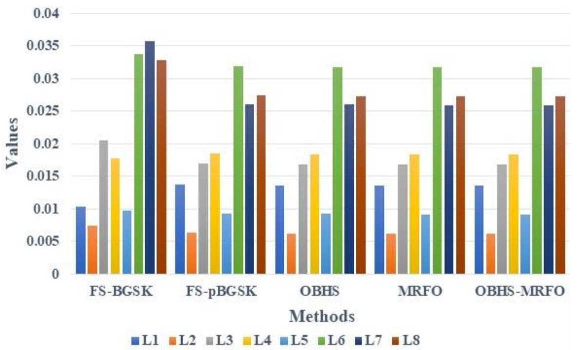
Table IV exhibits the average selected feature ratio to the number of total features evaluated on the datasets of L1–L8, M1–M8, and H1–H5. From Table IV, it is evident that OBHS-MRFO commendable results on H1–H5 dataset with values of about 0.39830, 0.41336, 0.36912, 0.36583, and 0.15783, respectively.

TABLE V. STANDARD DEVIATION OF FITNESS VALUES OF OPTIMIZERS

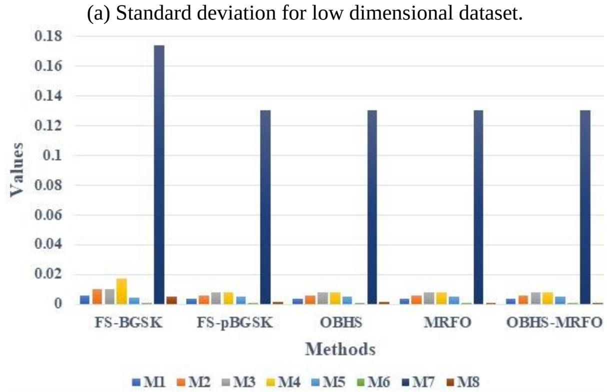
Datasets	FS-BGSK	FS-pBGSK	OBHS	MRFO	OBHS-MRFO
L1	0.0104	0.0137	0.01366	0.01365	0.01360
L2	0.0075	0.0063	0.00621	0.00617	0.00615
L3	0.0205	0.0169	0.01682	0.01681	0.01679
L4	0.0178	0.0185	0.01842	0.01841	0.01835
L5	0.0098	0.0093	0.00925	0.00919	0.00915
L6	0.0338	0.0319	0.03181	0.03175	0.03171
L7	0.0358	0.0261	0.02601	0.02593	0.02584
L8	0.0328	0.0274	0.02731	0.02729	0.02722
M1	0.0060	0.0036	0.00352	0.00344	0.00335
M2	0.0099	0.0056	0.00553	0.00550	0.00546
M3	0.0099	0.0082	0.00819	0.00812	0.00810
M4	0.0173	0.0078	0.00776	0.00771	0.00765
M5	0.0043	0.0053	0.00528	0.00520	0.00520
M6	0.0003	0.0001	0.00007	–0.00001	0.00002
M7	0.1740	0.1304	0.13031	0.13030	0.13029
M8	0.0049	0.0012	0.00117	0.00110	0.00108
H1	0.0293	0.0280	0.02799	0.02794	0.02992
H2	0.0152	0.0060	0.00599	0.00593	0.00589
H3	0.0050	0.0038	0.00380	0.00373	0.00366
H4	0.0287	0.0254	0.02534	0.02532	0.02524
H5	0.0016	0.0007	0.00066	0.00057	0.00051

Table V represents the standard deviation of optimizers, and Fig. 2 displays the standard deviations of L1–L8, M1–M8, and H1–H8 datasets, respectively.

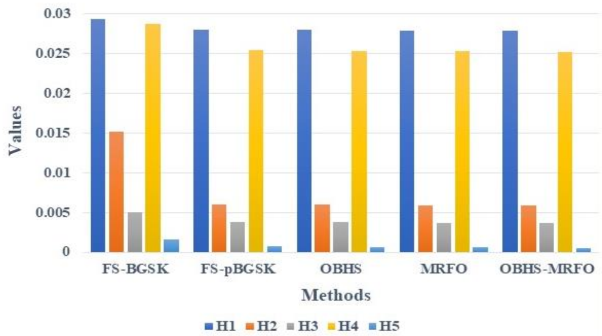
Table VI details the computation time consumed by various algorithms, while Fig. 3 represents the computation time on L1–L8, M1–M8, and H1–H8 datasets, respectively.



(a) Standard deviation for low dimensional dataset.

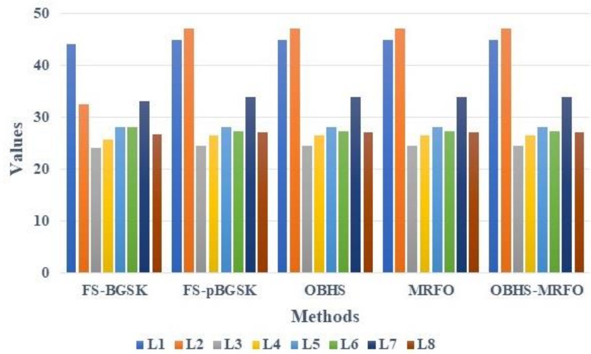


(b) Standard deviation for medium dimensional dataset.

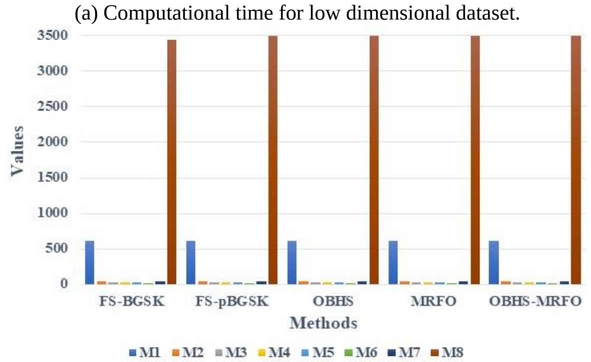


(c) Standard deviation for high dimensional dataset.

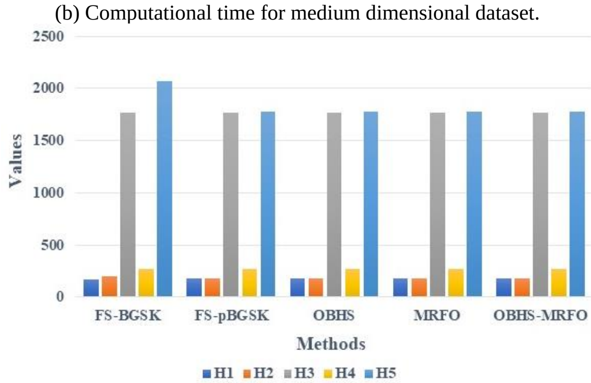
Fig. 2. Standard deviation.



(a) Computational time for low dimensional dataset.



(b) Computational time for medium dimensional dataset.



(c) Computational time for high dimensional dataset.

Fig. 3. Computational time.

It is also clearly proved from Tables II–VI, that if the dataset size is increased, the model's performance also increases.

TABLE VI. AVERAGE COMPUTATIONAL TIME FOR VARIOUS ALGORITHMS

Datasets	FS-BGSK	FS-pBGSK	OBHS	MRFO	OBHS-MRFO
L1	44.08	44.92	44.91991	44.91990	44.91989
L2	32.41	47.14	47.13999	47.13997	47.13989
L3	24.14	24.52	24.51998	24.51996	24.51988
L4	25.67	26.51	26.50992	26.50987	26.50977
L5	27.99	27.98	27.97991	27.97990	27.97989
L6	28.11	27.16	27.15992	27.15985	27.15979
L7	33.00	33.92	33.91997	33.91996	33.91991
L8	26.67	27.04	27.03991	27.03985	27.03977
M1	617.20	615.41	615.40995	615.40987	615.40980
M2	40.27	41.44	41.43999	41.43998	41.43993
M3	26.79	27.29	27.28995	27.28986	27.28984
M4	24.30	24.43	24.42990	24.42985	24.42983
M5	25.84	25.78	25.77992	25.77985	25.77977
M6	20.44	20.71	20.70996	20.70987	20.70980
M7	38.08	38.92	38.91993	38.91987	38.91978
M8	3441.45	3496.28	3496.27993	3496.27992	3496.27990
H1	172.18	178.32	178.31993	178.31986	177.31980
H2	201.69	184.02	184.01996	184.01989	184.01986
H3	1772.33	1767.84	1767.83995	1767.83990	1767.83987
H4	267.05	274.54	274.53995	274.53987	274.53978
H5	2073.50	1778.36	1778.35996	1778.35991	1778.35987

B. Comparative Analysis

This section presents a comparative analysis of the proposed model using performance metrics of average fitness function, accuracy, feature selection size, standard deviation, and computational time, as shown in Table VII. The existing models such as bGSK [22] and FS-pBGSK [23] are utilized to comparatively evaluate their ability of optimization. Table VII denotes comparison of OBHS-MRFO for H1 dataset. The OBHO-MRFO achieves average fitness function, accuracy, feature selection size, standard deviation, and computational time of 0.3995, 0.6274, 0.398, 0.0299, and 177.31, respectively.

TABLE VII. COMPARATIVE ANALYSIS USING H1 DATASET

Performance Metrics	Methods		
	bGSK [22]	FS-pBGSK [23]	Proposed OBHS-MRFO
Average fitness function	0.4151	0.3861	**0.3995**
Average accuracy	0.5825	0.6139	**0.6274**
Average selected feature ratio	N/A	0.38	**0.398**
Standard deviation	N/A	0.0280	**0.0299**
Computational time	N/A	178.32	**177.31**

C. Discussion

The bGSK [22] struggles to designate a suitable population size, whereas the FS-pBGSK [23] attains less accuracy with high computational time. Therefore, in the suggested framework, the opposition-based approach of OBHS enhances exploration, while MRFO's foraging behavior contributes to the efficient exploitation. However, as individual algorithms, both OBHS and MRFO have discrete limitations. OBHS suffers from premature convergence or slow convergence, and its effectiveness depends on the problem characteristics and parameter settings. MRFO, on the other hand, requires careful parameter tuning and does not perform optimally in certain situations. The attained result shows that the OBHS-MRFO achieves commendable results when the acquired average fitness function, accuracy, feature selection size, standard deviation, and computational time values of about 0.3995, 0.6274, 0.398, 0.0299, and 177.31, respectively, are contrasted against the values attained by bGSK and FS-pBGSK.

V. CONCLUSION

The motivation of the proposed feature selection is to determine an optimum subset feature from an initial dataset that enhances performance of ML by minimizing dimensionality and eliminating irrelevant features. The hybrid metaheuristic is the most common method for trading optimization problems. This manuscript proposes a feature selection methodology based on a hybrid of Opposition Based Harmony Search (OBHS) and Manta Ray Foraging Optimization (MRFO) to overcome the issues of minimum accuracy, which are produced by redundant and irreverent features. The University of California, Irvine (UCI) dataset that contains low, medium, and high dimensional datasets, is employed. Preprocessing is carried out along with min-max normalization that enhances the model's performance, besides the hybrid OBHS-MRFO being employed for feature selection. Then, the selected features are given as input to the classification process which includes KNN, ELM, MLP, and SVM. Finally, the performance of the presented model is evaluated and it achieves average fitness function value, average accuracy, average feature selection size, standard deviation, and computational time of 0.3995, 0.6274, 0.398, 0.0299, and 177.31, respectively, which are better outcomes when compared to those of bGSK [22] and FS-pBGSK [23]. In the future, this method will be extended to solve the data imbalance issues.

CONFLICTS OF INTEREST

The authors declare no conflict of interest.

AUTHOR CONTRIBUTIONS

The paper's conceptualization, methodology, software, validation, writing—original draft preparation, and project administration was handled by Thatikonda Somashekar, whereas formal analysis, investigation, resources, data curation, writing—review and editing, visualization, and supervision was handled by Srinivas Jagirdar. All authors had approved the final version.

REFERENCES

[1] N. L. S. Albashah and H. M. Rais, "Population initialization factor in binary multi-objective grey wolf optimization for features selection," *IEEE Access*, vol. 10, pp. 114942–114958, Oct 2022.

[2] A. Naskar, R. Pramanik, S. K. S. Hossain *et al.*, "Late acceptance hill climbing aided chaotic harmony search for feature selection: An empirical analysis on medical data," *Expert Syst. Appl.*, vol. 221, 119745, Jul. 2023.

[3] D. Moldovan, "Binary horse optimization algorithm for feature selection," *Algorithms*, vol. 15, no. 5, 156, May 2022.

[4] A. Shaddeli, F. S. Gharehchopogh, M. Masdari, and V. Solouk, "An improved African vulture optimization algorithm for feature selection problems and its application of sentiment analysis on movie reviews," *Big Data and Cognitive Computing*, vol. 6, no. 4, p. 104, Sep. 2022.

[5] M. K. Keleş and Ü. Kiliç, "Binary black widow optimization approach for feature selection," *IEEE Access*, vol. 10, pp. 95936–95948, Sep. 2022.

[6] Y. Chen, Z. Ye, B. Gao *et al.*, "A robust adaptive hierarchical learning crow search algorithm for feature selection," *Electronics*, vol. 12, no. 14, 3123, Jul. 2023.

[7] G.-L. Wang, S.-C. Chu, A.-Q. Tian *et al.*, "Improved binary grasshopper optimization algorithm for feature selection problem," *Entropy*, vol. 24, no. 6, 777, May 2022.

[8] A. A. Ewees, M. A. A. Al-qaness, L. Abualigah *et al.*, "Boosting arithmetic optimization algorithm with genetic algorithm operators for feature selection: Case study on cox proportional hazards model," *Mathematics*, vol. 9, no. 18, 2321, Sep. 2021.

[9] M. A. S. Ali, F. P. P. Rajeena, and D. S. A. Elminaam, "An efficient heap based optimizer algorithm for feature selection," *Mathematics*, vol. 10, no. 14, 2396, Jul. 2022.

[10] M. A. Elaziz, A. A. Ewees, M. A. A. Al-qaness *et al.*, "Feature selection for high dimensional datasets based on quantum-based dwarf mongoose optimization," *Mathematics*, vol. 10, no. 23, 4565, Dec. 2022.

[11] R. K. Eluri and N. Devarakonda, "Feature selection with a binary flamingo search algorithm and a genetic algorithm," *Multimedia Tools Appl.*, vol. 82, no. 17, pp. 26679–26730, Jul. 2023.

[12] M. Braik, "Enhanced Ali Baba and the forty thieves algorithm for feature selection," *Neural Comput. Appl.*, vol. 35, no. 8, pp. 6153–6184, Mar. 2023.

[13] J.-S. Pan, H.-J. Shi, S.-C. Chu *et al.*, "Parallel binary rafflesia optimization algorithm and its application in feature selection problem," *Symmetry*, vol. 15, no. 5, 1073, May 2023.

[14] E. Pashaei and E. Pashaei, "An efficient binary chimp optimization algorithm for feature selection in biomedical data classification," *Neural Comput. Appl.*, vol. 34, no. 8, pp. 6427–6451, Apr. 2022.

[15] L. Sun, H. Qin, K. Przystupa *et al.*, "A hybrid feature selection framework using improved sine cosine algorithm with metaheuristic techniques," *Energies*, vol. 15, no. 10, 3485, May 2022.

[16] H. Wu, S. Du, Y. Zhang *et al.*, "Threshold binary grey wolf optimizer based on multi-elite interaction for feature selection," *IEEE Access*, vol. 11, pp. 34332–34348, Mar. 2023.

[17] M. Xu, Q. Song, M. Xi, and Z. Zhou, "Binary arithmetic optimization algorithm for feature selection," *Soft Comput.*, vol. 27, no. 16, pp. 11395–11429, Aug. 2023.

[18] X. Wang, X. Dong, Y. Zhang, and H. Chen, "Crisscross Harris hawks optimizer for global tasks and feature selection," *J. Bionic Eng.*, vol. 20, no. 3, pp. 1153–1174, May 2023.

[19] L. Sun, Y. Xin, T. Chen, and B. Feng, "Rolling bearing fault feature selection method based on a clustering hybrid binary cuckoo search," *Electronics*, vol. 12, no. 2, 459, Jan. 2023.

[20] E. A. Zaimoğlu, N. Yurtay, H. Demirci, and Y. Yurtay, "A binary chaotic horse herd optimization algorithm for feature selection," *Eng. Sci. Technol. Int. J.*, vol. 44, 101453, Aug. 2023.

[21] L. Fang and X. Liang, "A novel method based on nonlinear binary grasshopper whale optimization algorithm for feature selection," *J. Bionic Eng.*, vol. 20, no. 1, pp. 237–252, Jan. 2023.

[22] P. Agrawal, T. Ganesh, D. Oliva, and A. W. Mohamed, "S-shaped and v-shaped gaining-sharing knowledge-based algorithm for feature selection," *Appl. Intell.*, vol. 52, no. 1, pp. 1–32, Jan. 2022.

[23] P. Agrawal, T. Ganesh, and A. W. Mohamed, "A novel binary gaining-sharing knowledge-based optimization algorithm for feature selection," *Neural Comput. Appl.*, vol. 33, no. 11, pp. 5989–6008, Jun. 2021.

[24] UCI-dataset. (2022). [Online]. Available: https://www.kaggle.com/datasets/mdwaquarazam/ucidatasetlist

[25] H. Kibriya, R. Amin, A. H. Alshehri *et al.*, "A novel and effective brain tumor classification model using deep feature fusion and famous machine learning classifiers," *Comput. Intell. Neurosci.*, vol. 2022, 7897669, Mar. 2022.

[26] A. Asokan, "A self-adaptable Manta ray optimized Gabor filter for satellite image enhancement," *Earth Sci. Inf.*, vol. 16, no. 2, pp. 1503–1517, Jun. 2023.

[27] M. Kusy and P. A. Kowalski, "Architecture reduction of a probabilistic neural network by merging k-means and k-nearest neighbour algorithms," *Appl. Soft Comput.*, vol. 128, 109387, Oct. 2022.

[28] L. Armi, E. Abbasi, and J. Zarepour-Ahmadabadi, "Texture images classification using improved local quinary pattern and mixture of ELM-based experts," *Neural Comput. Appl.*, vol. 34, no. 24, pp. 21583–21606, Dec. 2022.

[29] E.-S. M. El-Kenawy, S. Mirjalili, A. Ibrahim *et al.*, "Advanced meta-heuristics, convolutional neural networks, and feature selectors for efficient COVID-19 X-ray chest image classification," *IEEE Access*, vol. 9, pp. 36019–36037, Feb. 2021.

[30] M. A. Almaiah, O. Almomani, A. Alsaaidah *et al.*, "Performance investigation of principal component analysis for intrusion detection system using different support vector machine kernels," *Electronics*, vol. 11, no. 21, 3571, Nov. 2022.