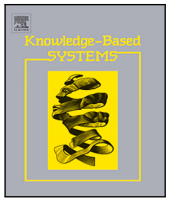
Knowledge-Based Systems 283 (2024) 111218

Contents lists available at ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys



An adaptive hybrid mutated differential evolution feature selection method for low and high-dimensional medical datasets

Reham R. Mostafa [a,b,*], Ahmed M. Khedr [c], Zaher Al Aghbari [c], Imad Afyouni [c], Ibrahim Kamel [d], Naveed Ahmed [c]

[a] Big Data Mining and Multimedia Research Group, Centre for Data Analytics and Cybersecurity (CDAC), Research Institute of Sciences and Engineering (RISE), University of Sharjah, Sharjah 27272, United Arab Emirates
[b] Information Systems Department, Faculty of Computers and Information Sciences, Mansoura University, Mansoura 35516, Egypt
[c] Department of Computer Science, University of Sharjah, Sharjah 27272, United Arab Emirates
[d] Department of Computer Engineering, University of Sharjah, Sharjah 27272, United Arab Emirates

ARTICLE INFO

Keywords:
Feature selection (FS)
Differential Evolution (DE)
Classification
Skin disease prediction

ABSTRACT

Feature selection (FS) constitutes a crucial endeavor in classification procedures, aiming to identify the minimal subset of features that maximizes classification accuracy. In the realm of combinatorial NP-hard challenges, FS assumes significance, prompting the utilization of robust metaheuristics as efficient wrapper-based FS strategies. However, the application of these wrapper metaheuristics to high-dimensional datasets, characterized by an abundance of features and a scarcity of samples, often results in a decline in effectiveness and escalated computational costs. Addressing these limitations, this study introduces the Adaptive Hybrid-Mutated Differential Evolution (A-HMDE) method, targeting the inherent drawbacks of the Differential Evolution (DE) algorithm. The A-HMDE incorporates four distinct strategies. Firstly, it integrates the mechanics of the Spider Wasp Optimization (SWO) algorithm into DE's mutation strategies, yielding enhanced performance marked by high accuracy and swift convergence towards global optima. Secondly, adaptive mechanisms are applied to key DE parameters, amplifying the efficiency of the search process. Thirdly, an adaptive mutation operator ensures a harmonious balance between global exploration and local exploitation during optimization. Lastly, the concept of Enhanced Solution Quality (ESQ), rooted in the RUN algorithm, guides DE to elude local optima, thus heightening the accuracy of obtained solutions. The efficiency of the A-HMDE approach is assessed by employing it as a FS method across diverse datasets encompassing the UCI repository, microarray data, and skin disease image dataset. The experimental results accentuate the method's remarkable ability to overcome challenges related to local minima and hasten the convergence process. A comprehensive comparison against contemporary cutting-edge algorithms highlights the considerable advancements achieved by the proposed method, with recorded accuracy ranging from 0.88 to 1.00.

1. Introduction

Medical data mining is the process of extracting valuable insights and patterns from large and complex medical datasets using various data mining and machine learning techniques [1]. It plays a crucial role in healthcare by enabling researchers and practitioners to analyze vast amounts of medical data to discover trends, correlations, and hidden information that can improve patient care, disease diagnosis, treatment outcomes, and medical research [2]. The challenge in medical data mining often involves filtering out irrelevant features from the data to improve the performance of classifiers and other analytical models [3]. This process helps in reducing noise and focusing on the most meaningful and predictive attributes, ultimately leading to more accurate results and insights. Therefore, feature selection is essential to ensure that the extracted patterns and insights are based on clinically relevant information.

Feature selection (FS) represents a critical step in the machine learning pipeline, enabling the construction of more accurate, efficient, and interpretable models, contributing to better healthcare outcomes [4]. FS offers several advantages, beginning with its ability to reduce data dimensionality [5]. This reduction facilitates faster and more efficient

* Corresponding author at: Big Data Mining and Multimedia Research Group, Centre for Data Analytics and Cybersecurity (CDAC), Research Institute of Sciences and Engineering (RISE), University of Sharjah, Sharjah 27272, United Arab Emirates.
E-mail addresses: REldeiasti@sharjah.ac.ae, reham_2006@mans.edu.eg (R.R. Mostafa), akhedr@sharjah.ac.ae (A.M. Khedr), zaher@sharjah.ac.ae (Z.A. Aghbari), iafyouni@sharjah.ac.ae (I. Afyouni), kamel@sharjah.ac.ae (I. Kamel), nahmed@sharjah.ac.ae (N. Ahmed).

https://doi.org/10.1016/j.knosys.2023.111218
Received 31 August 2023; Received in revised form 30 October 2023; Accepted 15 November 2023
Available online 21 November 2023

training of machine learning models by minimizing the number of features processed by the algorithm. Moreover, FS enhances model accuracy by eliminating irrelevant and redundant features that can introduce noise and bias to the data. Furthermore, FS contributes to model interpretability by facilitating a better understanding of the relationships between features and the outcome variable. Lastly, FS mitigates the risk of overfitting, which occurs when a model becomes overly complex and fits the training data too closely, resulting in poor performance on new data.

The FS, a binary optimization problem, entails the identification of a subset of informative features from an original set [6]. Due to the vast number of potential feature combinations, FS is considered NP-hard [7]. To tackle this challenge, several algorithms have been proposed, employing different methodologies. Filter methods evaluate features using statistical metrics like correlation or mutual information, selecting them based on relevance to the target variable. Wrapper methods, on the other hand, use a specific learning algorithm's performance to assess feature subsets, involving an iterative process of selection and evaluation. Hybrid methods strike a balance by combining aspects of both filter and wrapper techniques, aiming for computational efficiency while maintaining model performance [8,9].

Given the superior performance of the wrapper approach, several efficient FS algorithms have been developed using this method, categorized into three directions: exponential, sequential, and random search strategies [10,11]. The exponential method yields accurate outcomes at the cost of high computational demands [12]. In the sequential search, features are progressively added or removed from the subset, subsequently becoming fixed and unchangeable. This property can sometimes lead to being trapped in a local optimum [12]. Random search methods, on the other hand, employ randomness-based algorithms like simulated annealing, random generation, and metaheuristic algorithms to escape local optima [11].

Among these approaches, metaheuristic algorithms have garnered substantial attention due to their commendable ability to generate potential feature subsets [13]. Notably, various widely recognized metaheuristic algorithms have played a significant role in refining the wrapper approach. These include Equilibrium optimizer (EO) [14], Emperor Penguin Optimizer (EPO) [15], Butterfly Optimization Algorithm (BOA) [16], and Vortex Search Algorithm (VSA) [17]. Despite their popularity, many of these algorithms face challenges related to inadequate local and global search strategies, an imbalance in handling challenges, and low population diversity, collectively affecting their solution quality [18,19]. Consequently, the FS literature has witnessed a proliferation of modifications and hybrid approaches aimed at harnessing the advantages of various strategies and enhancing search efficiency. The aim is to "benefit from the best of both worlds". In this vein, various combinations of methods and search strategies have been proposed [12,17,20–23].

Evolutionary algorithms (EAs) have garnered widespread recognition as potent methods within the field of metaheuristics, renowned for their ability to tackle complex problems by offering robustness and global search capabilities. Among the array of EAs, differential evolution (DE) stands out as a particularly promising approach, celebrated for its adaptability and versatility. Introduced by Storn and Price in 1995 [24], DE operates on the principle of survival of the fittest, akin to genetic algorithm (GA) [25]. The algorithm has captivated the attention of researchers to tackle real-world optimization problems such as Robotics and Control Systems [26,27], Wireless Communication and Networks [28–30], supply chain and logistics [31–33], and environmental engineering and sustainable development [34–37]. This is primarily due to several competitive advantages it offers:

- Firstly, DE exhibits promising optimization abilities even with its simple structure. It outperforms other metaheuristic algorithms when faced with challenging problem features such as nonlinearity, multimodality, and nonseparability.
- Secondly, since 2005, multiple variants of DE have consistently secured high ranks in the Congress of Evolutionary Computation (CEC) competitions. This consistent performance underscores the capability of DE variants to effectively address a wide array of real-world applications, offering competitive levels of solution quality, convergence speed, and reliability.
- Lastly, DE possesses a desirable feature of low space complexity compared to other metaheuristic algorithms. This means that DE is highly scalable and capable of handling large-scale and computationally expensive optimization problems while requiring lower storage resources.

These strengths make DE a preferred choice for researchers and practitioners working on optimization problems in various domains. It has proven its effectiveness in addressing challenging optimization tasks and continues to be an active area of research and application.

The DE algorithm, a computational method that emulates the evolutionary dynamics of populations through iterative processes of population initialization, mutation, crossover, and selection, exhibits inherent limitations similar to other algorithms. These limitations include premature convergence, where the algorithm converges to suboptimal solutions prematurely, hindering exploration of the entire search space and restricting global optimum discovery. DE can also exhibit slow convergence, particularly in complex, high-dimensional problems, demanding a substantial number of iterations for an acceptable solution, thus incurring high computational costs. Additionally, DE is sensitive to control parameters like mutation and crossover rates; improper selection of these parameters can severely impact the algorithm's performance, potentially yielding unsatisfactory outcomes.

Researchers have been working on developing strategies to overcome these limitations. Various modifications, extensions, and hybridizations of the DE algorithm have been proposed, such as adaptive parameter control mechanisms [38,38,39], enhanced mutation and crossover strategies [40–42], and incorporation of local search techniques [43,44]. These endeavors aim to address the aforementioned limitations and augment the optimization capabilities of the DE algorithm across diverse domains, enabling it to navigate complex problem spaces and yield improved results effectively.

In this paper, an adaptive hybrid-mutated DE method is proposed, which encompasses four key strategies:

- Hybrid mutation strategies: The method introduces hybrid mutation strategies that combine the strengths of the SWO algorithm with DE mutation strategies. This integration enhances the performance of DE, enabling it to achieve the global optimal solution with high accuracy and rapid convergence.
- Adaptive control parameters: An adaptive approach is applied to the DE algorithm's crucial parameters, optimizing the search mechanism. This adaptability ensures that the algorithm can dynamically adjust its behavior to suit different problem landscapes and search spaces.
- Adaptive mutation operator: To strike a proper balance between global exploration and local exploitation, an adaptive mutation operator is employed. This operator intelligently adjusts the mutation process, allowing the algorithm to explore diverse regions while also fine-tuning its focus on promising areas.
- Enhanced Solution Quality (ESQ): The method leverages the concept of ESQ in the RUN algorithm to facilitate DE's escape from local optima. By improving the accuracy of the best solution obtained, ESQ enhances the algorithm's ability to overcome stagnation and find better solutions.

The subsequent sections of this paper are structured as follows: a survey of related research is presented in the section labeled "Related work". The methodologies employed in this study are elaborated upon in the "Background and algorithms" section. The introduction and detailed exposition of the novel Adaptive Hybrid Mutated Differential Evolution

(A-HMDE) approach is provided in the "Proposed method" section. Subsequently, the outcomes of the experiments and corresponding analyses are explicated in the "Experiments and analysis" section. Lastly, the conclusions of this study is encapsulated in the "Conclusion" section, where the key findings are succinctly summarized.

2. Literature review

This section provides a concise overview of notable advancements in DE algorithms specifically designed to tackle FS problems.

Ghosh et al. [45] introduced a self-adaptive DE (SADE) algorithm for hyperspectral image feature subset selection, addressing computational demands and redundancy. SADE integrated ReliefF to improve subset quality by removing duplicate features and employed a fuzzy k-nearest neighbor classifier for evaluation. Comparative analysis showcased SADE's superiority across three hyperspectral datasets, outperforming existing methods in terms of classification accuracy and Kappa coefficient. In a different context, Mlakar et al. [46] developed a Facial Expression Recognition (FER) system focusing on seven emotions. Their approach combined the histogram of oriented gradient descriptor (HOG) and difference feature vectors. To optimize FS, they introduced a modified multi-objective DE (DEMO), aiming to minimize features while improving SVM classifier accuracy. Statistical testing demonstrated the capability of the proposed method in selecting the most discriminative features of the different emotions, demonstrating efficiency. The FER system, coupled with this method, achieved exceptional emotion recognition rates and significantly reduced feature count, indicating its high performance and computational efficiency.

Vivekanandan et al. [47] introduced a novel FS method for predicting cardiovascular diseases. They proposed a modified version of the DE algorithm, specifically selecting the DE/rand/2/exp strategy from traditional DE methods. This strategy involved random vector selection for perturbation, aggregating weighted differences of four chosen vectors to create a mutant vector, and refining it through exponential crossover. Their hybrid model, combining fuzzy analytical hierarchy and feed-forward neural network techniques, outperformed existing methods with an 83% accuracy rate. Nayak et al. [48] introduced a new approach to FS for supervised classification using the Elitism-based multi-objective DE (EMODE) combined with the Extreme Learning Machine (ELM). The researchers aimed to optimize multiple objectives simultaneously, including the number of selected features, classification accuracy, and Minkowski score. They extensively compared their algorithm with classical and metaheuristic techniques across 21 benchmark datasets. The experimental results demonstrated the superior performance of their approach, outperforming alternative methods by a significant margin.

Yao et al. [49] presented two enhanced Binary DE (BDE) algorithms tailored for variable selection in developing reliable soft sensors for industrial applications. These methods effectively filtered out irrelevant variables and identified pertinent ones through a double-layer selection strategy, verified using root mean square error (RMSE). Experimental results substantiated the superiority of these algorithms over existing techniques, especially concerning prediction accuracy and stability. Hancer [50] introduced a multi-objective DE algorithm combined with filter criteria based on fuzzy and kernel measures. This approach aimed to improve the accuracy and interpretability of predictive models in diverse data domains. Through extensive experiments on various datasets, including discrete and continuous data types, the proposed criteria outperformed existing methods.

Zhang et al. [51] proposed a self-learning multi-objective FS approach using binary DE (BDE), known as MOFS-BDE. MOFS-BDE combined self-learning capabilities, innovative mutation schemes, and efficient selection strategies that allow optimizing classification accuracy and feature subset size simultaneously. MOFS-BDE demonstrated remarkable efficacy, achieving substantial enhancements in classification accuracy while retaining a minimal set of chosen features, as confirmed through extensive experimentation and analysis. Rivera-Lopez et al. [52] introduced a permutational-based DE algorithm for flexible feature subset selection, enhancing FS performance and classifier accuracy. The algorithm optimizes subset size and population diversity, demonstrated through effective compact feature subset generation and high classifier accuracy in experiments.

Pan et al. [53] introduced CMODE, a competitive mechanism-based multi-objective DE algorithm focusing on FS. By integrating competitive interactions, CMODE balanced convergence and diversity, showing effectiveness in multi-objective optimization and generating competitive feature subsets across benchmarks. Hancer et al. [54] introduced a solution for cost-sensitive classification in evolutionary computation-based FS. They integrated a fuzzy mutual estimator criterion into the DE framework, creating a specialized filter for cost-sensitive FS. Extensive experiments across various benchmarks proved its effectiveness, reducing error rates and feature costs by eliminating irrelevant features within a reasonable timeframe.

Hu et al. proposed NetG-DE and NetG-MODE, innovative FS algorithms utilizing network science to uncover feature relationships. These methods outperformed in high-dimensional data, showcasing their potential for significant FS enhancement through intricate node and cluster analysis. Hancer et al. [55] introduced an evolutionary filter FS method integrating Neighborhood Component Analysis (NCA) into DE. It aims for optimal feature subsets, maximizing class separation and minimizing dimensionality. Extensive experiments exhibited its superiority over recent evolutionary, information-theoretic, and rough set-based methods in both single- and multi-objective scenarios. Agrawal et al. [56] introduced FSSDE, a multimodal multiobjective DE algorithm for FS. They proposed a unique initialization method, maintaining feature diversity and enhancing exploration. A balanced niching technique ensured effective space exploitation, while an archive method maximized exploration. Empirical results demonstrated FSSDE's effectiveness in identifying diverse feature subsets with consistent classification accuracy across datasets.

Table 1 provides a comprehensive summary of the mentioned research papers, highlighting their focus on the DE and its various modified versions, which have been published in recent years.

3. Background and algorithms

3.1. Feature selection problem

As previously noted, the challenge of FS is rooted in its nature as a multi-objective optimization problem. The fundamental objective entails the selection of an optimal subset of features that maximizes accuracy or minimizes error. This multi-objective fitness function can be reformulated into a single-objective function with the following structure:

$$\text{Fitness} = \alpha \times \text{Classification Error} + \beta \times \frac{\text{Number of Selected Features}}{\text{Total Features}} \tag{1}$$

In this context, the ratio of selected features to the overall number of features embodies the FS rate. It is essential to highlight that parameters α and β are confined within the interval [0,1], with the additional constraint that $\beta = \alpha - 1$. The classifier used in this study is k-nearest neighbors (kNN).

3.1.1. k-nearest neighbors (kNN)

The kNN algorithm is a simple and effective machine learning algorithm that is often used for classification tasks [58]. The kNN algorithm has been used successfully in a variety of domains, including free text, images, audio, and video. It is a popular choice for many data science projects because it is relatively easy to understand and implement. It works by finding the k most similar points to a new point

Table 1
The summary of DE variants for FS.

Paper	Year	DE variant	Main strategies used	Merits	Applications
[45]	2013	Self-adaptive DE (SADE)	SADE generates subsets using ReliefF and assesses them with fuzzy k-nearest neighbor classifiers to eliminate redundant features	self-adaptive control is employed to optimize parameter values, striking a balance between exploration and exploitation.	Hyperspectral images datasets
[46]	2017	Multi-objective DE algorithm (MODE)	FER approach using HOG descriptors, feature vector differences, SVM, and modified multi-objective DE FS method.	Improve accuracy, overcome local solutions	Facial Expression Recognition (FER) system
[47]	2017	Modified DE (MDE) algorithm	MDE employed DE/rand/2-wt/exp strategy: random vector perturbation, four-vector selection, weighted differences for mutant vector, and exponential crossover.	Enhance global solution finding, accuracy, and reduce computation time	Cardiovascular disease
[48]	2018	Elitism-based Multi-objective DE (EMODE) algorithm	EMODE incorporates the widely recognized Elitism principle from NSGA-II into its selection process.	Effectively exploit and avoid local solutions.	Selecting the most informative features for different benchmark datasets
[49]	2018	Binary DE (BDE) algorithms	(1) Parallel BDE (PBDE) uses parallel short evolutionary paths with an "early stopping" strategy. (2) Boosting BDE (BBDE) combines boosting technique with BDE for variable selection	Improve convergence speed, and avoid local solutions effectively.	Selecting variables for nonlinear soft sensor development
[50]	2019	MODE filter method	MODE, based on information theory and rough set theory, is useful for continuous datasets	Improves prediction model accuracy and interpretability	Selecting benchmark datasets' most informative features
[51]	2020	Self-learning multi-objective binary DE (MOFS-BDE) algorithm	MOFS-BDE uses self-learning capabilities, innovative mutation schemes, and efficient selection strategies	Expedite the convergence speed and enhance the capability to evade local optima	Selecting the most informative features for different benchmarks datasets
[52]	2020	Permutational-based DE algorithm	Permutation-based strategy helps the algorithm to effectively explore the feature space	Boost population diversity, overcome local minima, and enhance accuracy.	Selecting relevant features from high-dimensional datasets
[53]	2022	Competitive mechanism based multi-objective DE (CMODE) algorithm	Integrating competitive interactions into the evolutionary process	Enhances exploration and avoids local solutions	Selecting the most informative features for different benchmarks datasets
[54]	2022	Cost-sensitive DE FS method (CDE)	Apply information theory and fuzzy mutual estimator to DE	It allows efficient exploration of feature space within a reasonable time	Selecting informative features for diverse UCI benchmark datasets
[57]	2023	Two network-based DEs, namely, NetG-DE and NetG-MODE	Discover hidden relationship patterns within the feature space using network science techniques	The feature network reveals hidden patterns not visible in the original space, emphasizing its structural features	Selecting benchmark datasets' most informative features
[55]	2023	Single-objective (DENCA) and multiobjective (MODENCA) DE scenarios	Integrating an NCA-inspired goal function into the DE framework	NCE helps to find a feature subset that optimizes class separation and minimizes dimensionality	Selecting the most informative features for different benchmarks datasets
[56]	2023	Multimodal multiobjective DE algorithm (FSSDE)	Using (1) probability initialization, (2) niching technique, and (3) convergence archive	These techniques preserve diversity among selected features, aiding in comprehensive search space exploration	Selecting the most informative features for different benchmarks datasets

in the training dataset and then assigning the new point to the same class as the majority of its k nearest neighbors. The performance of the kNN algorithm can be affected by several factors, including the choice of distance metric and the value of the k parameter. The Euclidean distance is the most commonly used distance metric, calculated using Eq. (2), but other metrics can also be used. The value of k is a hyper-parameter that controls the number of neighbors that are considered when classifying a new point.

$$d(y, y_i) = \sqrt{\sum_{k=1}^{n} (y_j - y_{ij})^2} \tag{2}$$

Algorithm 1 demonstrates the kNN classification steps.

Algorithm 1 kNN pseudocode.

Inputs: Import the train and test data.
Outputs: Put the test point in the appropriate class.
Pick the value of k for each data point in the test set.
while the terminal condition is not met **do**
  Determine the Euclidean distance using Eq. (2).
  Maintain a list of the Euclidean distances and arrange them.
  Take the first k values.
end while
Return *Accuracy*.

3.2. Differential evolution (DE)

DE is a population-based metaheuristic optimization algorithm inspired by Darwin's theory of evolution [24]. It was first introduced by Storn and Price in 1997 and has since been widely used in various optimization and engineering domains. DE works by iteratively generating new candidate solutions, called offspring, by combining existing solutions in the population. The three fundamental mechanisms of DE are mutation, crossover, and selection.

1. Initialization: The DE algorithm generates X potential solutions to start. Randomly produced solutions provide the algorithm

with a diversified search starting point. During the tth iteration, each DE solution i is represented as $X_i^t = \left(X_{i,1}, X_{i,2}, \ldots, X_{i,Dim}\right)$, where i ranges from 1 to N, and N is the population size, while the dimension of the problem being optimized is Dim.
The location of the ith solution is X_i. The randomization technique for the first solution positions is as follows:

$$X_i = X_{\min} + \text{rand}[0,1]\left(X_{\max} - X_{\min}\right) \tag{3}$$

where $X_{\min}$ and $X_{\max}$ represent the lower and upper solution search space bounds, respectively. Uniform distribution rand$[0,1]$ generates random real values between 0 and 1. DE commonly randomizes solution position setting to promote diversity and exploration.

2. Mutation: Mutation, in biological terms, refers to the sudden alteration of characteristics observed in a gene of a chromosome. In the context of evolutionary computation, mutation involves a random perturbation process applied to selected decision variables. According to DE, a mutant vector Y_i is created by mutating a target vector X_i.
The DE mutation strategy is usually written as 'DE/*/n', where n is the number of difference vectors and * is the target vector. The five most popular mutation techniques in DE, each with its own unique search processes, are as follows:

$$\text{DE/rand/1}: \quad Y_i^t = X_{r_1}^t + F\left(X_{r_2}^t - X_{r_3}^t\right) \tag{4}$$

$$\text{DE/rand/2}: \quad Y_i^t = X_{r_1}^t + F\left(X_{r_2}^t - X_{r_3}^t\right) + F\left(X_{r_4}^t - X_{r_5}^t\right) \tag{5}$$

$$\text{DE/ best /1}: \quad Y_i^t = X_{best}^t + F\left(X_{r_1}^t - X_{r_2}^t\right) \tag{6}$$

$$DE/\text{ best /2}: \quad Y_i^t = X_{best}^t + F\left(X_{r_1}^t - X_{r_2}^t\right) + F\left(X_{r_3}^t - X_{r_4}^t\right) \tag{7}$$

$$\text{DE/ current-to-best /1}: \quad Y_i^t = X_i^t + F\left(X_{best}^t - X_i^t\right) + F\left(X_{r_1}^t - X_{r_2}^t\right) \tag{8}$$

In DE mutation, multiple population indicators are used to build the mutant vector. r_1 is the index of the DE solution chosen as the base vector, whereas r_2, r_3, r_4, and r_5 are the indices of randomly selected DE solutions used to generate the mutant vector. It is vital to notice that r_1, r_2, r_3, r_4, and r_5 must be unique integers in the range $[1, N]$. During mutation, X_{best}^t selects the best individual solution in the DE population as the target vector during the mutation process. The mutation procedure is guided by this target vector.
The scaling factor, F, controls the mutation process. It is a real-valued parameter between 0 and 1. Choosing the right F value is crucial since it balances algorithm exploration and exploitation searches. A well-chosen F value prevents premature convergence or poor convergence pace, enabling efficient optimization.

3. Crossover: The crossover step probabilistically combines the mutant vector and target vector to form a trial vector, which represents the offspring solution. The mutant solution gives the target solution traits via this crossover process. Uniform and exponential crossover are typical DE crossover operators. A crossover rate (CR) controls the probabilistic combination of components in the uniform crossover scheme. The crossover rate, between 0 and 1, influences the possibility of a mutant vector component being chosen for the trial vector. The trial solution generated through uniform crossover can be mathematically defined as follows:

$$Z_{i,j}^t = \begin{cases} Y_{i,j}^t & \text{if } rand(0,1) \le CR \text{ or } j = k \\ X_{i,j}^t & \text{Otherwise} \end{cases} \tag{9}$$

where $randi, j$ is a random number and $k \in \{1, 2, \ldots, Dim\}$ is a randomly selected dimension index to ensure that the trial solution Z_i inherits at least one-dimensional component from the mutant vector Y_i.

4. Selection: The selection process in DE is critical in choosing which solutions, whether target or trial, will survive and be included in the next search iteration X_i^{t+1}. This mechanism maintains the population size in each generation. DE's selection process is mathematically expressed as follows:

$$X_i^{t+1} = \begin{cases} Z_i^t & \text{if } f\left(Z_i^t\right) \le f\left(X_i^t\right) \\ X_i^t & \text{Otherwise} \end{cases} \tag{10}$$

DE selects solutions by assessing their fitness using the objective function $f(.)$. If the newest trial vector Z_i^t yields a superior fitness value than the current target vector X_i^t, it replaces it in the next iteration. By keeping and passing on the best solutions to the following generations, the selection process aids in the algorithm's refinement and convergence.

After creating the next generation's population, mutation, crossover, and selection continue to be applied. DE uses this iterative cycle to explore the solution space, enhance solution quality, and find the optimal or near-optimal optimization problem solution.

3.3. Spider wasp optimizer (SWO)

Spider Wasp Optimizer (SWO) takes its inspiration from the hunting strategies of female wasps that seek out spiders for egg-laying [59]. These wasps exhibit a multi-step process: identifying suitable spiders, immobilizing and transporting them to prepared nests, and ultimately depositing an egg on the spider's abdomen before sealing the nest. By simulating the search, selection, and placement behaviors of the spider wasps, the SWO algorithm is designed to tackle complex optimization problems with a high degree of precision and adaptability. The SWO algorithm starts by creating an initial population, which comprises a collection of individual solutions representing potential solutions to the optimization problem. Subsequently, it employs a mathematical representation of spider-wasp behavior to navigate the solution space and identify the most optimal solution.

- Searching stage (exploration): This stage mirrors female wasps finding the finest spiders to lay their eggs. The female wasp randomly searches for a spider for her offspring. In Eq. (11), female wasps explore by updating their location with a constant velocity at each generation t.

$$\overrightarrow{SW}_i^{t+1} = \overrightarrow{SW}_i^t + \mu_1 * \left(\overrightarrow{SW}_a^t - \overrightarrow{SW}_b^t\right) \tag{11}$$

where a and b are two random indices from the population to identify female wasps' exploratory orientation, and μ_1 is utilized to determine continued movement throughout the current direction using the formula:

$$\mu_1 = |rn| * r_1 \tag{12}$$

where r_1 is a random number between zero and one and rn is a normal distribution-generated random number.
If they lose a spider from the orb, female wasps seek the vicinity. SWO employs a secondary equation featuring distinct exploration strategies to probe the region surrounding the displaced spider using a smaller step size as in Eq. (13). This equation updates the current female wasp with a constant motion at each generation based on the location of a randomly selected female wasp representing the fallen spider. This equation:

$$\overrightarrow{SW}_i^{t+1} = \overrightarrow{SW}_c^t + \mu_2 * \left(\vec{L} + \vec{r_2} * (\vec{H} - \vec{L})\right) \tag{13}$$

$$\mu_2 = B * \cos(2\pi l) \tag{14}$$

$$B = \frac{1}{1+e^{l}} \tag{15}$$

where c is a population random index and l is an integer between 1 and −2. Finally, the female wasp's next position is randomly chosen between Eqs. (11) and (13) as follows:

$$\overrightarrow{SW}_i^{t+1} = \begin{cases} \text{Eq. (11)} & r_3 < r_4 \\ \text{Eq. (13)} & \text{otherwise} \end{cases} \tag{16}$$

where r_3 and r_4 are random numbers in [0, 1].

- Following and escaping stage: Spider wasps attack prey in the web hub, but they sometimes fall to the ground. The wasp paralyzes and drags the fallen spiders to its nest. If the spider falls too quickly, the wasp has to grab it while escaping. This behavior captures two trends: The wasp hunts spiders using Eq. (17) to adjust its position and chase prey. It also evades other wasps by increasing distance with each iteration, allowing prey to hide. The first trend, expressed in Eq. (17), models wasp chasing. The initial distance between prey and wasp is small and changes based on their speeds. This equation covers two cases: (1) wasp faster than prey $C > 0.5$, and (2) prey faster than wasp $C < 0.5$, where C controls wasp's speed, starting at two and reducing linearly to zero. When wasp is faster than 0.5, distance between it and prey increases, simulating escape. If wasp's speed is under 0.5, its position changes very little, affecting its ability to reach prey.

$$\overrightarrow{SW}_i^{t+1} = \overrightarrow{SW}_i^{t} + C * \left| 2 * \vec{r}_5 * \overrightarrow{SW}_a^{t} - \overrightarrow{SW}_i^{t} \right| \tag{17}$$

$$C = \left(2 - 2 * \left(\frac{t}{t_{\max}}\right)\right) * r_6 \tag{18}$$

where a is a randomly generated population index, t and t_max are the current and maximum iterations, r_5 is a vector of random values in [0, 1], and r_6 is a random number.
As the spider escapes the female wasp, their distance grows. Exploitation is barely starting. Exploitation turns into exploration as journey distances increase. This formula illustrates this progression:

$$\overrightarrow{SW}_i^{t+1} = \overrightarrow{SW}_i^{t} * \overrightarrow{vc} \tag{19}$$

where $\overrightarrow{vc}$ is a vector from the normal distribution between k and $-k$. The value of k is calculated using Eq. (20) to gradually increase the distance between the female wasp and the spider.

$$k = 1 - \left(\frac{t}{t_{\max}}\right) \tag{20}$$

The following equation shows how those two trends tradeoff randomly:

$$\overrightarrow{SW}_i^{t+1} = \begin{cases} \text{Eq. (17)} & r_3 < r_4 \\ \text{Eq. (19)} & \text{otherwise} \end{cases} \tag{21}$$

- Nesting behavior (exploitation): Wasp females transport paralyzed spiders into their nests. Spider wasps construct cells in soil, mud nests in various materials, or use existing nests or cavities. Given these diverse nesting methods, SWO employs two equations to simulate these behaviors. The first equation, represented in (22), guides the spider towards the best location, identified as the optimal spot for creating a nest to house the paralyzed spider and lay an egg on its abdomen. The second equation, represented in (23), constructs nests based on the position of a randomly selected female spider from the population. This is done with an additional step size to prevent multiple nests from forming in the same spot.

$$\overrightarrow{SW}_i^{t+1} = \overrightarrow{SW}^{*} + \cos(2\pi l) * \left(\overrightarrow{SW}^{*} - \overrightarrow{SW}_i^{t}\right) \tag{22}$$

where SW^* represents the best-known solution.

$$\overrightarrow{SW}_i^{t+1} = \overrightarrow{SW}_a^{t} + r_3 * |\gamma| * \left(\overrightarrow{SW}_a^{t} - \overrightarrow{SW}_i^{t}\right) + (1 - r_3) * \vec{U} * \left(\overrightarrow{SW}_b^{t} - \overrightarrow{SW}_c^{t}\right) \tag{23}$$

where r_3 is a random number between [0, 1] and γ is the levy flight number. a, b, and c are an index of three randomly chosen population solutions, and binary vector $\vec{U}$, calculated as in Eq. (24) determines the step size to prevent creating two nests in the same location.

$$\vec{U} = \begin{cases} 1 & \vec{r_4} > \vec{r_5} \\ 0 & \text{otherwise} \end{cases} \tag{24}$$

where $\vec{r_4}$ and $\vec{r_5}$ are random vectors between 0 and 1. Randomly swapping (22) and (23) using the formula:

$$\overrightarrow{SW}_i^{t+1} = \begin{cases} \text{Eq. (22)} & r_3 < r_4 \\ \text{Eq. (23)} & \text{otherwise} \end{cases} \tag{25}$$

4. Proposed adaptive hybrid mutated DE (A-HMDE) method

This section presents a comprehensive description of the proposed A-HMDE method. It starts by discussing the limitations of the original DE algorithm, highlighting the areas that need improvement. Subsequently, the framework of the A-HMDE method is introduced, outlining its key features and innovations aimed at addressing the identified limitations and enhancing the optimization process.

4.1. Shortcomings of original DE algorithm

The DE algorithm, like other computational methods based on evolutionary dynamics, has inherent limitations. These limitations include premature convergence, where the algorithm settles on suboptimal solutions early without fully exploring the search space, leading to less-than-optimal results. DE can also exhibit slow convergence rates, particularly in complex and high-dimensional problems, requiring many iterations to find acceptable solutions, making it computationally expensive. Additionally, DE's performance is sensitive to parameter selection, such as mutation and crossover rates, and suboptimal choices can lead to poor outcomes.

4.2. Framework of the proposed A-HMDE method

To overcome the limitations mentioned earlier, this paper presents an adaptive hybrid-mutated DE (A-HMDE) approach. The A-HMDE approach incorporates four key strategies to enhance the performance of the DE algorithm. Firstly, it integrates the mechanism of the SWO algorithm into DE's mutation strategies, significantly improving the algorithm's ability to find global optimal solutions with high accuracy and rapid convergence. Secondly, an adaptive parameter strategy optimizes DE's key parameters, improving its overall effectiveness. Thirdly, an adaptive mutation operator is employed to balance between global exploration and local exploitation throughout the optimization process. Lastly, the concept of Enhanced Solution Quality (ESQ) is leveraged to aid DE in escaping local optima, ultimately enhancing the accuracy of the acquired best solution. Each of these four strategies will be elaborated upon in the subsequent subsections.

4.2.1. Hybrid mutation strategies

This section introduces a novel hybridization approach, leveraging the mechanisms of the SWO to enhance the performance of DE mutation strategies. The primary objective of this hybridization is to synergistically combine DE's mutation strategies with SWO's biologically inspired exploration and exploitation mechanisms, creating a robust optimization technique for FS problem. To improve the searchability and convergence speed of the DE algorithm, a robust mutation operator is crucial. However, DE's capacity for global and local search has limitations. To address this, the study introduces two mutation strategies, DE/rand-global-WSO/2 and DE/rand-local-WSO/2, inspired by SWO's mechanisms. These strategies aim to enhance the conventional DE/rand/2 and DE/best/2 methods, boosting both global and local search capabilities and convergence rates. The formulations of these novel strategies are presented below.

- DE/rand-global-WSO/2: The main goal of the proposed DE/rand-global-WSO/2 strategy is to enhance global search efficiency by integrating the WSO exploration mechanism into the DE algorithm. In the DE framework, two key mutation strategies, DE/rand/1 and DE/rand/2, are employed. DE/rand/1 selects three individuals and mutates one to create a trial vector, promoting exploration but potentially resulting in slow convergence and premature convergence due to limited information. On the other hand, DE/rand/2 uses two difference vectors, enabling broader exploration, particularly in rugged terrains. While this approach can identify promising regions more quickly, it may overfit to local optima in smoother landscapes, risking premature convergence.

  With the aim of enhancing the exploration capabilities of the DE/rand/2 strategy, this study introduces a new mutation approach strategy DE/rand-global-WSO/2. This strategy is developed by incorporating the WSO global search step into the DE/rand/2 mutation process, resulting in the following modified mutant strategy:

$$Y_i^t = X_{r_1}^t + F\left(X_{r_2}^t - X_{r_3}^t\right) + \mu_2 * \left(L + X_{r_4}^t * (H - L)\right) \tag{26}$$

  The expression "$\mu_2 * \left(L + X_{r_4}^t * (H - L)\right)$" is adapted from the exploration equation of the WSO (Eq. (13)). This term influences the agent's position update by introducing a consistent movement at each generation. This movement is determined by the position of a randomly selected solution from the population.
- DE/rand-local-WSO/2: The focus of the "DE/rand-local-WSO/2" strategy is to enhance local search by integrating WSO's exploitation mechanism into DE. Within the DE framework, the widely used mutation strategies DE/best/1 and DE/best/2 are designed to exploit promising regions in the search space. DE/best/1 involves mutating the global best individual with a single randomly chosen difference vector, making it computationally efficient but potentially leading to premature convergence, especially in intricate and multimodal landscapes. On the other hand, DE/best/2 utilizes two difference vectors and the best individual to generate a trial vector, aiming for a balance between exploration and exploitation of the best solution.

  The second modification aims to enhance the exploitation capacity of the DE/rand/2 strategy through the introduction of a new mutation strategy termed DE/rand-local-WSO/2. This approach incorporates the element of the local search step from the WSO method into the DE/best/2 mutation strategy. The modified mutant strategy is shown as follows:

$$Y_i^t = X_{\text{best}}^t + F\left(X_{r_1}^t - X_{r_2}^t\right) + \cos(2\pi l) * \left(X_{\text{best}}^t - X_i^t\right) \tag{27}$$

  The expression "$\cos(2\pi l) * \left(X_{\text{best}}^t - X_i^t\right)$" is derived from the exploitation equation of the WSO (Eq. (22)). This term is designed to attract the agent towards the region containing the most suitable solution. Notably, the mutation strategy effectively exploits the nearby region of each X_i^t in the direction of the solution X_{best}^t, significantly enhancing the algorithm's search capability, particularly in the later stages of the evolutionary process. As a result, the algorithm becomes more adept at reaching the global optimal solution for the given optimization problem.

4.2.2. Adaptive control parameters

The scale factor (F) and crossover rate (CR) are two crucial control parameters in the DE algorithm, exerting a significant impact on its overall performance. These parameters are instrumental in striking the right balance between exploration and exploitation during the optimization process, thereby influencing the algorithm's convergence speed and the quality of solutions obtained. To achieve a well-balanced optimization approach that emphasizes global exploration in the early stages and local exploitation in the later stages, this paper uses an adaptive mutation crossover operator. This operator dynamically adjusts the values of F and CR based on the optimization progress, ensuring that the algorithm can efficiently explore the search space during the initial iterations while fine-tuning its focus on promising regions to improve convergence as the optimization proceeds.

- Adaptive scale factor (F): The performance of the DE algorithm is significantly influenced by the choice of the scale factor (F), which balances exploration and exploitation during optimization. Dynamic adjustment of the F during the optimization process is highlighted as a means to achieve the desired balance. In the early optimization stages, a higher F brings benefits such as enhanced exploration, enabling trial vectors to deviate more from parent vectors, thereby fostering a diverse search across the solution space. This maintains population diversity, preventing premature convergence to local optima. As optimization progresses, transitioning to a smaller F promotes exploitation and faster convergence. A smaller F restricts trial vector deviation, focusing the search around identified promising regions. This enhances exploitation and speeds up convergence by concentrating the search around these regions, thus facilitating more efficient solution refinement.

  To strike a balance between global search in the early stages and local optimization in the later stages, this paper uses an adaptively adjusting scale operator. This operator allows DE to dynamically change the values of F based on the fitness values of the optimal offspring and parent individuals during the iterative process. The expression is described as follows:

$$F_i = \begin{cases} F_{i-1} & f_{i\_next} \le f_i \\ F_0 + k * \tau & f_{i\_next} > f_i \quad \text{and} \quad \tau < 0.5 \\ 1 - k * \tau & \text{else} \end{cases} \tag{28}$$

  where k is a fixed value within the range (0, 1), τ is the ratio of the current iteration count to the maximum iteration count, and F_0 is the initial value of the scaling factor.
- Adaptive crossover rate (CR)

  Moreover, the performance of the DE algorithm is profoundly influenced by the proper selection of the crossover rate (CR). The CR is a crucial control parameter that governs the probability of recombination between the parent vector and the trial vector during the mutation process. It plays a vital role in determining the balance between exploration and exploitation in the optimization process. The impact of the crossover rate (CR) on the performance of the DE algorithm is noteworthy. Firstly, CR significantly affects information exchange between the parent and trial vectors during the mutation process. A higher crossover rate increases the probability of recombination, promoting extensive information sharing and exploration of promising features. Conversely, a lower CR restricts information exchange, relying more on mutation for exploration. Secondly, the choice of CR influences the algorithm's convergence speed. Higher CR values can accelerate convergence by facilitating efficient information sharing of valuable information among individuals, but excessively high values might cause premature convergence or overexploitation, limiting the algorithm's ability to escape local optima. Therefore, selecting an appropriate CR is crucial, as it directly impacts both the exploration–exploitation balance and the algorithm's convergence rate in DE.

  To strike a balance between global search and local search, this study introduces an adaptive adjustment mechanism for CR. This operator dynamically adapts CR based on the fitness values of parent and offspring individuals, achieving a well-balanced optimization process. The expression for the adaptively adjusting

crossover rate is as follows:

$$CR_i = \begin{cases} CR_{i-1} & f_{i\_next} \le f_i \\ CR_0 + k * \tau & f_{i\_next} > f_i \quad \text{and} \quad \tau < 0.5 \\ 1 - k * \tau & \text{else} \end{cases} \tag{29}$$

where CR_0 is the initial value of the crossover rate.

4.2.3. Adaptive mutation operators

The effectiveness of the DE algorithm is significantly impacted by the selection of mutation strategies, which in turn affects convergence, population diversity, and exploration capabilities. To tackle issues like premature convergence and stagnation, this paper uses an adaptive mutation operator denoted as $\rho^{(t)}$. This operator dynamically regulates the algorithm's abilities for global exploration and local exploitation. The formulation of the adaptive mutation operator $\rho^{(t)}$ is as follows:

$$\rho^{(t)} = \left| \frac{f_{best}^{(t)} - f_{mean}^{(t)}}{f_{best}^{(t)} - f_{worst}^{(t)}} \right| \tag{30}$$

where $f_{mean}^{(t)}$, $f_{best}^{(t)}$ and $f_{worst}^{(t)}$ are the mean, best, and worst objective function values of individuals in the previous generation, respectively.

The adaptive parameter $\rho^{(t)}$ takes values in the range [0,1] and plays a crucial role in identifying the evolutionary stages during the search process. In the early stage of the optimization process, when $\rho^{(t)}$ is close to 1, it indicates that the population is far from the region of the global optimum. At this point, the focus is on global exploration in the search space. This approach enhances exploration and helps in locating the region of the global optimum. As the optimization process progresses and the value of $\rho^{(t)}$ approaches 0, it signifies that the population is getting closer to the region of the global optimum. This corresponds to the later stage of the search, where the emphasis shifts towards local exploitation. This adjustment aims to strengthen the exploitation ability of the algorithm and further improve the quality of solutions found.

4.2.4. Enhanced Solution Quality (ESQ)

The RUNge Kutta Optimizer (RUN) is an innovative metaheuristic algorithm inspired by the mathematical principles of the Runge Kutta (RK) method, a well-known mathematical technique [60]. For a comprehensive understanding of the RUN algorithm's mathematical background, please refer to [60]. One of the key strengths of RUN lies in its Enhanced Solution Quality (ESQ) feature, which leverages the most recent leading solution to enhance solution quality and accelerate convergence. The RUN algorithm uses ESQ to guarantee that each solution continually progresses towards the best solution. The ESQ method takes the best solution (x_best) and the average of the three random solutions (x_avg) to build a new solution (x_new1). The following scheme outlines the process for creating the solution (x_{new2}) using ESQ:

If $rand < 0.5$

  If $w < 1$

$$x_{new2} = x_{new1} + r \times w \times \left| (x_{new1} - x_{avg}) + randn \right| \tag{31}$$

  Else

$$x_{new2} = (x_{new1} - x_{avg}) + r \times w \times \left| (x_{new1} - x_{avg}) + randn \right|$$

  End

End

where

$$x_{new1} = \beta \times x_{avg} + (1 - \beta) \times x_{best} \tag{32}$$

$$x_{avg} = \frac{x_{r1} + x_{r2} + x_{r3}}{3} \tag{33}$$

$$w = \text{rand}(0, 2) \times \exp\left(-c\left(\frac{i}{Maxi}\right)\right) \tag{34}$$

In these equations, β is a random number between 0 and 1, while c represents a random number equivalent to $(5 \times rand)$, and w denotes a random number that decreases with increasing iterations. r is an integer number that can take the values of 1, 0, or −1. Here, x_{best} refers to the best solution.

It is important to highlight that the solution computed in this step as (x_{new2}) may not necessarily possess a superior fitness value compared to the current solution. In order to offer another opportunity to generate a better solution, a new solution (x_{new2}) is produced with the subsequent definition:

If $rand < w$

$$x_{new3} = (x_{new2} - rand \times x_{new2}) + SF \times (rand \times x_{RK} + (v \times x_b - x_{new2})) \tag{35}$$

End

Here, v is a random number with a value of $2 \times rand$, and SF is calculated using Eq. (36). Also, x_{RK} is defined by Eq. (38). The primary objective of Eq. (35) is to update and improve the position of the solution xnew2, moving it towards a better position in the search space.

$$SF = 2 \times \left(\frac{1}{2} - rand\right) \times f \tag{36}$$

where f is defined by using the following Eq. (37)

$$f = a \times \exp\left(-b \times \text{rand} \times \left(\frac{i}{Maxi}\right)\right) \tag{37}$$

where a and b are constants, and i represents the number of iterations, and $Maxi$ is the maximum number of iterations.

$$x_{RK} = k_1 + 2 \times K_2 + 2 \times K_3 + K_4 \tag{38}$$

where the three coefficients (k2, k3, and k4) are defined as:

$$K_1 = \frac{1}{2\Delta x} \times (rand \times x_w - y \times x_b) \tag{39}$$

$$y = \text{round}(1 + rand) \times (1 - rand) \tag{40}$$

$$K_2 = \frac{1}{2\Delta x} \left(rand \times (x_w + r_1 \times K_1 \times \Delta x) - (y \times x_b + r_2 \times K_1 \times \Delta x)\right) \tag{41}$$

$$K_3 = \frac{1}{2\Delta x} \left(rand \times \left(x_w + r_1 \times \left(\frac{1}{2}K_2\right) \times \Delta x\right) - \left(y \times x_b + r_2 \times \left(\frac{1}{2}K_2\right) \times \Delta x\right)\right) \tag{42}$$

$$K_4 = \frac{1}{2\Delta x} \left(rand \times (x_w + r_1 \times K_3 \times \Delta x) - (y * x_b + r_2 \times K_3 \times \Delta x)\right) \tag{43}$$

where r_1 and r_2 are two random numbers in $[0, 1]$, x_b denotes the best x, x_w denotes worst x. Furthermore, Δx is defined as $= 2 \times rand \times | Step |$, where $Step$ is calculated as $rand \times \left(\left(x_b - rand \times x_{avg}\right) + \gamma\right)$. The scale factor γ is determined by $rand \times \left(x_n - \text{rand} \times \left(y - L_{bb}\right)\right) \times \exp\left(-4 \times \frac{i}{Max_i}\right)$, where L_{bb} represents the solution space's lower pound. The new solution (x_{new3}) is implemented when the condition $rand < w$ is met. The primary objective of Eq. (35) is to move the solution x_{new2} towards a better position.

4.3. Model of A-HMDE method

The presented A-HMDE method, depicted in Fig. 1 and Pseudocode 2, aims to enhance the performance of the traditional DE algorithm through the incorporation of four strategic approaches, effectively improving its exploration and exploitation capabilities. In this section, we delve into the main steps of the proposed A-HMDE method.

Step 1 (Initialization).

- The key parameters of the A-HMDE algorithm are initialized, which include the population size, the maximum number of generations, lower and upper bounds, initial scale factor (F_0), and initial crossover rate (CR_0).

- An initial population (X_i) consisting of N agents are generated uniformly at random using Eq. (3).

Step 2 (Mutation). The mutation step comprises exploration and exploitation phases.

- To enhance exploration capabilities, we introduce the use of the original DE/rand/1 and proposed DE/rand-global-WSO/2 mutation strategies, as demonstrated in Eqs. (4) and (26), respectively. Additionally, an adaptive scale factor F_i, given in Eq. (28), is introduced to replace the conventional scale factor F.
- For exploitation, we incorporate the original DE/best/1 and proposed DE/rand-local-WSO/2 mutation strategies, illustrated in Eqs. (6) and (27), respectively, along with the adaptive scale factor F_i.
- To determine the mutation strategies during the search process, an adaptive mutation operator $\rho^{(t)}$ is utilized, as depicted in Eq. (30). During the early stage, when $\rho^{(t)}$ is close to 1, indicating the population is distant from the global optimum region, DE/rand/1 and DE/SWO-global/2 are randomly chosen with a probability of 0.5, facilitating exploration to identify the global optimum region. During the later stage, when $\rho^{(t)}$ approaches 0, signifying the population is close to the global optimum region, DE/best/1 and DE/SWO-local/2 are randomly selected with a probability of 0.5, enabling enhanced exploitation and solution refinement.

Step 3 (Crossover). The crossover operator, presented in Eq. (9), is applied to the population. A higher crossover rate encourages a greater contribution of the mutant vector Y_i^t to the trial vector Z_i^t, enhancing population diversity and global search. Conversely, a lower crossover rate retains the previous state X_i^t with a higher probability, refining the trial vector for improved solution quality. To address these aspects, an adaptive crossover rate, defined in Eq. (29), is incorporated into the proposed A-HMDE method.

Step 4 (Selection). The selection operator, presented in Eq. (10), is employed to select individuals for the next generation based on their fitness values.

Step 5 (Enhanced Solution Quality (ESQ)). In order to elevate the overall solution quality and mitigate the risk of converging to local optima during each iteration, the A-HMDE method incorporates an ESQ strategy. The implementation of ESQ ensures that each solution undergoes a refinement process to move towards a better position in the search space. This refinement is conditionally triggered based on a comparison with a random value ($rand < 0.5$), where $rand$ represents a random number between 0 and 1. If the newly generated solution, x_{new2}, based on Eq. (31), does not yield satisfactory results compared to the current solution, an additional solution, x_{new3}, is generated using Eq. (35) in an attempt to achieve an improved outcome.

Algorithm 2 Pseudocode of A-HMDE method.

Input: Population size (N_p), maximum number of iterations (T_{max}), dimension of problem (d), and objective function $f()$
Output: Optimal solution
Generate the initial population of random solutions X_i using Eq. (3)
Initiate F_0 and CR_0
for $t = 1 : T_{max}$ **do**
  for $i = 1 : N_p$ **do**
    Compute the adaptive parameters F_i and CR_i using Eq. (28) and (29), respectively.
    Compute $\rho^{(t)}$ using Eq. (30)
    if $\rho^{(t)} > 0.5$ **then** // Exploration phase //
      if $rand > 0.5$ **then**
        Apply DE/rand/1 mutation strategy using Eq. (4)
      else
        Apply DE/rand-global-WSO/2 mutation strategy using Eq. (26)
      end if
    else if $\rho^{(t)} < 0.5$ **then** // Exploitation phase //
      if $rand > 0.5$ **then**
        Apply DE/best/1 mutation strategy using Eq. (6)
      else
        Apply DE/rand-local-WSO/2 mutation strategy using Eq. (27)
      end if
    end if
    Execute crossover operation using Eq. (9)
    Apply the selection operator using Eq. (10), consider both the offspring and the parent's fitness levels, and keep the fittest one x_i.
    Initialize ESQ with the current best solution
    if $rand < 0.5$ **then**
      Compute solution x_{inew2} using Eq. (31)
      if $f(x_i) < f(x_{inew2})$ **then**
        if $rand < w$ **then**
          Compute solution x_{new3} using Eq. (35)
        end if
      end if
    end if
  end for
  Update solution x_{best}
  $k++$
end for
Return best solution x_{best}.

5. Experiments and analysis

Within this section, an exhaustive comparative analysis is conducted to evaluate the performance of the A-HMDE methodology against several contemporary state-of-the-art techniques. This evaluation pertains to the task of selecting optimal subsets of features across various medical datasets. The study encompasses two essential evaluations: FS utilizing benchmark datasets and a real-world application involving the prediction of skin diseases. All experimental procedures were executed on MATLAB 2022b, utilizing an Intel® Core™ i7 CPU operating at 3.40 GHz, accompanied by 16 GB of RAM, and running the Microsoft Windows 11 operating system.

Several metaheuristics were evaluated and compared with the proposed A-HMDE in this experiment to ensure a fair assessment. The selected metaheuristics include Tasmanian devil optimization (TDO) [61], Moth-flame optimization (MFO) [62], Gorilla troops optimizer (GTO) [63], Tunicate swarm algorithm (TSA) [64], Particle swarm optimization (PSO) [65], Fick's law algorithm (FLA) [66], genetic algorithm (GA) [25], SWO [59], improved gorilla troops optimizer (mGTO) [19], improved African vulture optimization algorithm (Sin-Cos-bIAVOA) [67], and Improved bald eagle search (mBES) [68] . All algorithms were subjected to uniform evaluation conditions involving 30 search agents and a maximum of 100 iterations. To nullify the influence of stochastic initialization, the experiments were conducted across 30 independent runs. The dimensions of all algorithms were aligned with the original dataset's feature count, thereby ensuring uniformity and equitable comparison.

5.1. Experimental series 1: Feature selection on benchmark datasets

This section is dedicated to the comprehensive evaluation of the proposed methodology by means of a comparative analysis against other established FS techniques. The primary aim is to discern the efficacy and supremacy of the proposed approach in terms of its ability to select pertinent features when contrasted with alternative methods.

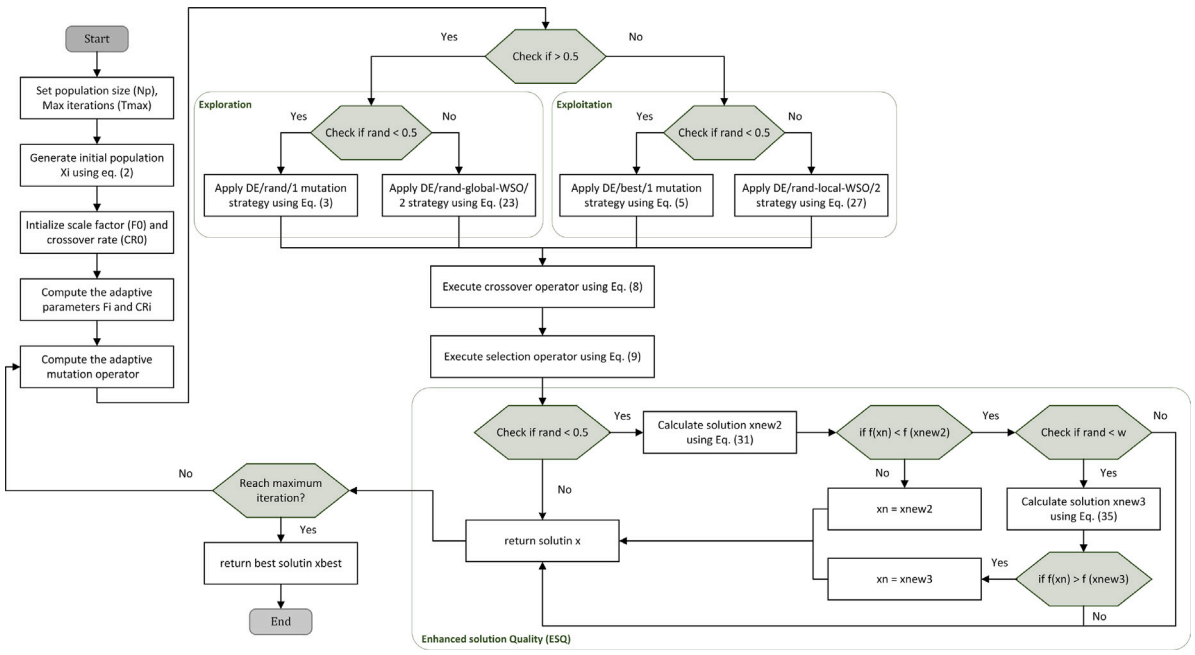


Fig. 1. Flowchart of the proposed A-HMDE method.

5.1.1. A-HMDE method for FS

This section offers a comprehensive elucidation of the implementation process of the proposed A-HMDE method for FS. The process encompasses four distinct steps, each elaborated upon as follows:

- Step 1: The initiation of the A-HMDE method involves the generation of a population comprising N individuals, where each individual embodies a potential feature subset for the given dataset. This population, denoted as X_i, is generated using the formulation presented in Eq. (3). Within this equation, lb_i and ub_i, respectively, denote the lower and upper boundaries of the specific problem. For this study, lb_i is assigned a value of 0, while ub_i is set to 1. Additionally, for the binary representation of each solution X_i, a conversion is executed using the following equation:

$$BX_i = \begin{cases} 1 & \text{if } X_i > 0.5 \\ 0 & \text{otherwise} \end{cases} \tag{44}$$

- Step 2: This step is centered on assessing the quality of each binary solution BX_i by computing the objective function according to Eq. (1). Within this equation, the values $\alpha = 0.99$ and $\beta = 1 - \alpha$ are adroitly incorporated to strike an equilibrium between the error engendered by KNN learning algorithm and the count of features selected through the optimization process.
- Step 3: During this stage, the principal process of updating solutions in A-HMDE is conducted. The revised candidate solutions are subsequently assessed through the utilization of the fitness function (1), leading to the identification of the most optimal solution. This sequence is recurrently performed until the set termination conditions are satisfied, which, in the scope of this study, is defined by the attainment of the maximum number of iterations.
- Step 4: In the classification Step, the feature subset selected by the A-HMDE method is applied to the testing dataset. The performance evaluation of the proposed approach is then conducted, encompassing various performance metrics.

5.1.2. Dataset description and performance criteria

To validate the efficacy of the proposed A-HMDE algorithm, a collection of eleven datasets sourced from the UCI repository [69] was employed. These datasets exhibit diverse levels of dimensionality, falling into categories of low and high dimensionality. Additionally, experimentation was conducted on six publicly available microarray datasets. These datasets encompass gene expression data associated with various diseases such as Colon cancer, Lung cancer, Ovarian cancer, Central Nervous System (CNS) disorders, and leukemia, among others. Each dataset consists of a significant number of features (genes) and a very small number of samples. Table 2 offers a brief summary of these datasets. The objective of this validation is to systematically test the performance and effectiveness of the A-HMDE method across several dataset dimensions in order to provide a thorough evaluation of its capabilities.

The classification technique employs the Hold-out strategy, which involves randomly dividing the dataset into two segments: 80% for the training set and 20% for the testing set. In addition, the selected classifier is the KNN algorithm, which utilizes the Euclidean distance metric. The value of the parameter K has been set to 5. The analysis of results for FS on sixteen datasets involves the utilization of various evaluation metrics. These metrics include mean and standard deviation values for accuracy, fitness value, sensitivity, specificity, feature selection size, CPU time, and the application of the Friedman test [70]. The purpose of these analyses is to statistically examine the experimental outcomes and determine the performance and ranking of the algorithms.

5.1.3. Results & discussion of A-HMDE on benchmark datasets

This subsection aims to compare the performance of the proposed A-HMDE method with other MHs across multiple evaluation criteria.

- Regarding fitness, which gauges the efficiency of methods in minimizing the fitness value, Table 3 showcases the outcomes of the compared methods, presenting the mean and standard deviation of the fitness function. The proposed A-HMDE method demonstrated superiority over other established techniques, achieving the best outcomes in 75% of the datasets (12 out of 16), with TDO leading in only three datasets. These findings strongly highlight the effectiveness of A-HMDE across a range of FS tasks. The same trend is visible in the standard deviation metric, where A-HMDE achieved notably lower standard deviation values, often less than 0.04 and close to zero. This signifies the stability of A-HMDE's search strategy, which strikes a balance between exploration and

Table 2
Datasets.

Dataset	# Features	# Instances	# Classes	Disease type	Source of datasets
Low dimensional datasets					
Arrhythmia	279	452	16	Cardiac arrhythmia	UCI repository
BreastEW	30	596	2	Breast Cancer	
Cleveland	13	297	5	Heart disease	
Diabetes	8	768	2	Diabetes	
HeartEW	13	270	2	Heart disease	
Hepatitis	18	79	2	Hepatitis	
Lymphography	18	148	4	Lymph disease	
Parkinson	22	194	2	Parkinsons	
High dimensional datasets					
Retionpathy	19	1151	2	Diabetic Retinopathy	UCI repository
Prostate	10 509	102	2	Prostate cancer	
Thyroid	21	7199	3	Thyroid	
CNS	7129	60	2	Central Nervous System (CNS)	Micoarray datasets
Colon	2000	60	2	Colon cancer	
Leukemia	7129	72	2	Acute myeloid leukemia (AML)	
Lung Cancer	12 533	181	2	Lung cancer	
Ovarian	15 154	253	2	Ovarian cancer	
SRBCT	2308	83	4	Small Round Blue Cell Tumors (SRBCT)	

Table 3
Fitness value comparison between A-HMDE and other optimization methods.

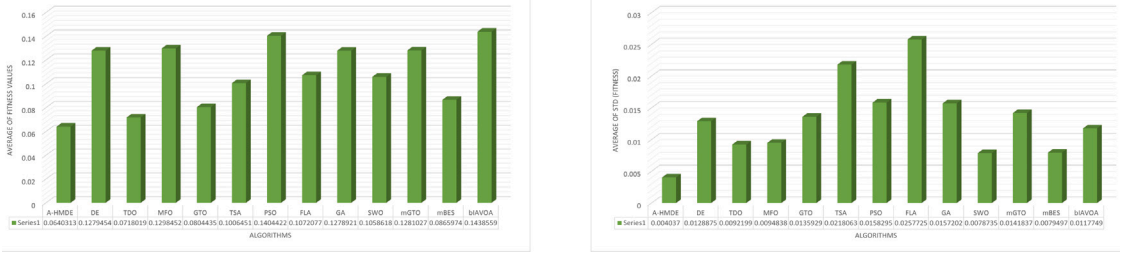
Dataset	Measures	A-HMDE	DE	TDO	MFO	GTO	TSA	PSO	FLA	GA	SWO	mGTO	mBES	bIAVOA
Low dimensional datasets														
Arrhythmia	Mean	0.204909	0.27822	0.220187	0.289574	0.266603	0.29571	0.298906	0.30706	0.25683323	0.33976	0.25766	**0.18943**	0.32389
	Std	0.034061	0.01743	0.032187	0.0166	0.030956	0.03124	0.026913	0.03975	0.01526155	**0.00659**	0.01437	0.00957	0.01612
BreastEW	Mean	0.020411	0.0215	**0.019805**	0.020827	0.027883	0.03467	0.0283	0.0321	0.02700548	0.0255	0.02154	0.02008	0.03034
	Std	**0.001996**	0.00288	0.003693	0.003274	0.004328	0.00466	0.007205	0.00547	0.0040522	0.00413	0.00497	0.00308	0.0032
Diabetes	Mean	0.209096	0.2099	**0.208214**	0.2099	**0.208214**	0.21247	0.212779	0.21607	0.23517857	0.21324	0.23526	0.23518	0.23533
	Std	0.003527	0.00461	8.6E−17	0.004611	8.6E−17	0.00652	0.007016	0.00718	**2.8666E−17**	0.00234	0.00031	2.9E−17	0.00043
HeartEW	Mean	0.080529	0.08087	**0.078285**	0.078862	0.079343	0.09938	0.085737	0.09109	0.13084936	0.10708	0.12751	0.12507	0.13157
	Std	0.00533	0.00556	**0.00056**	0.000577	0.000982	0.00989	0.008708	0.00793	0.0130241	0.01724	0.00898	0.01149	0.00663
Hepatitis	Mean	**0.17706**	0.1825	0.184813	0.190022	0.190801	0.20062	0.219666	0.21322	0.22621286	0.18791	0.17801	0.18877	0.20308
	Std	0.019618	0.0486	0.014047	0.034373	0.013917	0.01821	0.040441	0.03975	0.03903917	0.02156	0.03239	**0.01074**	0.01441
Lymphography	Mean	**0.068867**	0.07498	0.070481	0.07294	0.08537	0.11454	0.097993	0.10124	0.09379215	0.07135	0.07842	0.08514	0.07615
	Std	**2.57E−05**	0.01071	0.01325	0.014015	0.016431	0.02849	0.028529	0.03013	0.01826582	0.02073	0.0117	0.01418	0.00644
Parkinson	Mean	**0.026294**	0.04692	0.029438	0.038986	0.030997	0.04663	0.047515	0.04213	0.10270323	0.07982	0.10245	0.10245	0.10401
	Std	**0**	0.01023	0.008505	0.012935	0.010036	0.01009	0.010394	0.01249	0.00028598	0.02435	4.3E−17	4.3E−17	0.00072
High dimensional datasets														
Retionpathy	Mean	**0.212632**	0.23461	0.229859	0.228952	0.237561	0.25489	0.243111	0.25098	0.21916541	0.25128	0.21462	0.2136	0.22947
	Std	**0**	0.00932	0.004741	0.005494	0.00963	0.00914	0.012383	0.00977	0.00460573	0.0123	0.00172	0.00156	0.00612
Prostate	Mean	**2.14E−06**	0.09304	4.76E−06	0.099135	2.3E−05	0.01179	0.098807	0.04866	0.10124883	0.0474	0.08964	0.01886	0.15301
	Std	**4.4E−07**	0.01661	4.1E−06	4.43E−05	2.25E−05	0.02182	7.5E−05	0.04485	0.0267363	0.01489	0.01983	0.02434	0.01927
Thyroid	Mean	**0.008097**	0.0094	**0.008097**	**0.008097**	0.008942	0.01183	0.011942	0.01005	0.02294065	0.02109	0.02187	0.02231	0.02681
	Std	**1.9E−18**	0.00202	**1.9E−18**	**1.9E−18**	0.001848	0.00343	0.003467	0.00157	0.00089336	0.00167	0.00045	0.00061	0.00117
CNS	Mean	**0.016578**	0.252	0.07432	0.252245	0.082547	0.16501	0.268426	0.1741	0.09263049	0.08741	0.12837	0.08562	0.15875
	Std	**2.84E−05**	5.2E−05	0.026102	4.29E−05	0.038886	0.055	0.034719	0.06132	0.02604382	4.4E−05	0.04344	0.00021	0.02612
Colon	Mean	**1.65E−05**	0.24773	0.022871	0.248185	0.038102	0.08379	0.277992	0.1068	0.44279981	0.15704	0.45304	0.09903	0.40479
	Std	**5.3E−06**	0.04805	0.036775	0.032003	0.040128	0.06667	0.039362	0.06439	0.03202243	3.8E−05	0.02399	0.05139	0.03635
Leukemia	Mean	**2.1E−06**	0.1364	7.71E−06	0.136724	0.013212	0.03301	0.136405	0.07285	0.0215952	0.07091	0.03091	2.9E−06	0.06796
	Std	**7.39E−07**	4.2E−05	9.19E−06	4.74E−05	0.027826	0.03478	4.4E−05	0.05805	0.03173232	2.6E−05	0.03402	1.2E−06	0.0016
Lung Cancer	Mean	**2.38E−06**	0.07689	0.002435	0.077212	0.009699	0.02657	0.076921	0.01943	0.03306683	0.0049	0.0529	6.9E−06	0.04678
	Std	**3.74E−07**	7.7E−05	0.007634	1.7E−05	0.012448	0.02114	4.83E−05	0.01539	0.01148502	1.4E−05	4.8E−05	4.9E−06	0.00765
Ovarian	Mean	**1.85E−06**	0.06293	2.84E−06	0.068924	0.007772	0.01359	0.068637	0.02927	0.02062715	0.02431	0.02396	1.9E−06	0.02009
	Std	**5.21E−07**	3.9E−05	1.68E−06	0.009342	0.010027	0.00938	0.009377	0.01389	9.4135E−05	1.8E−05	3.1E−05	5.8E−07	0.00018
SRBCT	Mean	**4.33E−06**	0.03926	1.08E−05	0.056938	2.69E−05	0.00584	0.073937	0.00027	0.01962483	0.00481	0.03349	1E−05	0.08964
	Std	**0**	0.02997	1.1E−05	0.018366	2.16E−05	0.01841	0.02459	0.00043	0.02798057	5.4E−05	0.03068	5E−06	0.042
Friedman's mean rank		1.5625	7.53125	2.59375	7.65625	4.90625	7.9375	10.1875	8.625	8.96875	7.5625	7.96875	5.1875	10.3125
Rank		1	5	2	7	3	8	12	10	11	6	9	4	13

exploitation. Fig. 2 illustrates the average fitness results for all methods, where A-HMDE achieved the best average fitness compared to the other methods. Moreover, the low standard deviation values indicate the stability of A-HMDE across multiple runs, solidifying its strength in addressing different FS problems.

- In terms of accuracy, the mean correct classification rates derived from A-HMDE and the other methods are documented in Table 4. It is evident that A-HMDE outperformed other methods in twelve out of sixteen datasets, whereas TDO demonstrated better results in just three datasets and achieved comparable performance to A-HMDE in four datasets. In comparison to the typical DE method, A-HMDE demonstrated improved performance. It is worth mentioning that A-HMDE achieved the best level of accuracy in all nine high-dimensional datasets that were employed. On the other hand, TDO yielded identical findings to A-HMDE in four of the high-dimensional datasets. Fig. 3 presents the average accuracy results for all methods. Among the methods, A-HMDE has the greatest average accuracy. The experimental findings highlight the ability of the proposed A-HMDE method to identify the most informative features with higher accuracy values. The robustness of A-HMDE is further evidenced by its low standard deviation across most datasets, as shown in Fig. 3.
- In terms of sensitivity, Table 5 compares the results of the A-HMDE with other methods under the same conditions. A-HMDE achieved the highest sensitivity value in thirteen out of sixteen datasets, followed by TDO and MFO, which obtained the highest

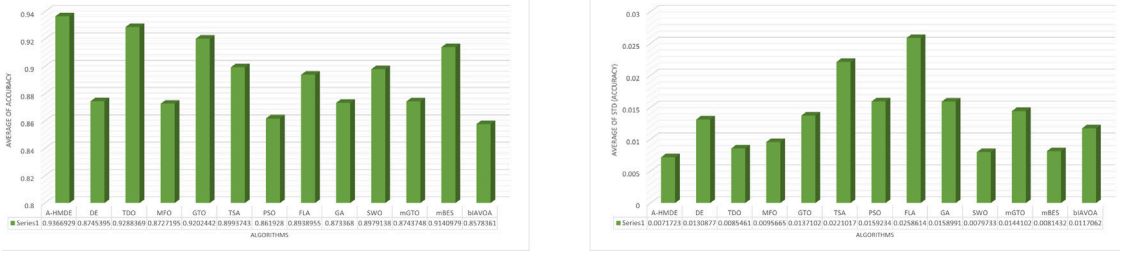
Table 4
Accuracy comparison between A-HMDE and other optimization methods.

Dataset	Measures	A-HMDE	DE	TDO	MFO	GTO	TSA	PSO	FLA	GA	SWO	mGTO	mBES	bIAVOA
Low dimensional datasets														
Arrhythmia	Mean	0.794469	0.72368	0.77823	0.71214	0.731403	0.701583	0.7019236	0.69446	0.74379	0.66188	0.7444	**0.80967**	0.67551
	Std	0.033536	0.01753	0.03229	0.01668	0.031191	0.031603	0.0270705	0.03874	0.01531	**0.0067**	0.01456	0.00969	0.01544
BreastEW	Mean	0.981908	0.98136	**0.983**	0.98246	0.975329	0.967105	0.9747807	0.97149	0.97643	0.97917	0.98191	0.98246	0.97533
	Std	**0.002193**	0.003	0.00388	0.0032	0.004771	0.005064	0.0070722	0.00599	0.0042	0.00439	0.00503	0.0032	0.00354
Diabetes	Mean	0.791396	0.79058	**0.79221**	0.79058	**0.792208**	0.788149	0.7881494	0.7849	0.76623	0.7914	0.76623	0.76623	0.76623
	Std	0.003247	0.00444	**0**	0.00444	**0**	0.006217	0.0062171	0.00665	2.3E−16	0.00222	2.3E−16	2.3E−16	2.3E−16
HeartEW	Mean	0.923611	0.92361	**0.92593**	**0.92593**	**0.925926**	0.902778	0.9189815	0.91435	0.87269	0.89699	0.87616	0.87847	0.87269
	Std	0.006325	0.00633	**2.3E−16**	**2.3E−16**	**2.29E−16**	0.010692	0.0092593	0.00926	0.01331	0.01785	0.00887	0.01165	0.00633
Hepatitis	Mean	**0.822581**	0.81855	0.81452	0.81048	0.808468	0.798387	0.7802419	0.78629	0.77419	0.81452	**0.82258**	0.81048	0.79839
	Std	0.020402	0.0498	0.01443	0.03509	0.014275	0.018624	0.0411738	0.04059	0.03907	0.02204	0.03332	**0.01102**	0.01443
Lymphography	Mean	**0.933804**	0.92802	0.93175	0.92973	0.91717	0.886718	0.9049808	0.90279	0.90891	0.93305	0.92475	0.91684	0.92894
	Std	0.013918	0.01099	**0.0011**	0.014	0.01658	0.028718	0.0289097	0.03081	0.01942	0.0211	0.01232	0.01537	0.00774
Parkinson	Mean	**0.974359**	0.95353	0.97115	0.96154	0.969551	0.953526	0.9535256	0.95833	0.89744	0.92308	0.89744	0.89744	0.89744
	Std	**0**	0.01034	0.00876	0.01324	0.010336	0.010336	0.0103362	0.01282	2.3E−16	0.02477	2.3E−16	2.3E−16	2.3E−16
High dimensional datasets														
Retionpathy	Mean	**0.787879**	0.76677	0.77137	0.77246	0.763528	0.74513	0.7583874	0.75027	0.78139	0.75065	0.78528	0.78658	0.77273
	Std	**1.17E−16**	0.00934	0.00505	0.00591	0.009856	0.0092	0.0122666	0.00945	0.00421	0.01193	0.00224	0.00209	0.00586
Prostate	Mean	**1**	0.91071	**1**	0.90476	**1**	0.988095	0.9047619	0.95238	0.9	0.95714	0.91429	0.98095	0.84762
	Std	**0**	0.01684	**0**	1.2E−16	**0**	0.022043	1.187E−16	0.04409	0.02703	0.01506	0.02008	0.02459	0.02008
Thyroid	Mean	**0.993746**	0.99259	**0.99375**	**0.99375**	0.993052	0.98957	0.9905027	0.99201	0.97962	0.98221	0.98027	0.97977	0.97739
	Std	**1.22E−16**	0.00179	**1.2E−16**	**1.2E−16**	0.001699	0.003653	0.0033366	0.00137	0.00066	0.00152	0.00048	0.00076	0.00107
CNS	Mean	**0.983333**	0.75	0.925	0.75	0.916667	0.833333	0.7333333	0.825	0.90833	0.91667	0.875	0.91667	0.84167
	Std	0.035136	**0**	0.02635	**0**	0.039284	0.055556	0.0351364	0.06149	0.02635	0	0.04392	**0**	0.02635
Colon	Mean	**1**	0.75385	0.97692	0.75385	0.961538	0.915385	0.7230769	0.89231	0.55385	0.84615	0.54615	0.9	0.59231
	Std	**0**	0.04865	0.03716	0.03243	0.040542	0.067353	0.0397229	0.06487	0.03243	1.2E−16	0.02433	0.05192	0.03716
Leukemia	Mean	**1**	0.86667	**1**	0.86667	0.986667	0.966667	0.8666667	0.92667	0.98	0.93333	0.97333	**1**	0.93333
	Std	**0**	2.3E−16	**0**	2.3E−16	0.028109	0.035136	2.341E−16	0.05837	0.0322	**0**	0.03443	**0**	**0**
Lung Cancer	Mean	**1**	0.92683	0.99756	0.92683	0.990244	0.973171	0.9268293	0.98049	0.96829	1	0.95122	**1**	0.95366
	Std	**0**	1.2E−16	0.00771	1.2E−16	0.012595	0.021356	1.17E−16	0.01543	0.01178	**0**	**0**	**0**	0.00771
Ovarian	Mean	**1**	0.94118	**1**	0.93529	0.992157	0.986275	0.9352941	0.97059	0.98039	0.98039	0.98039	**1**	0.98039
	Std	**0**	**0**	**0**	0.00947	0.010125	0.009471	0.0094715	0.01386	1.2E−16	1.2E−16	1.2E−16	**0**	1.2E−16
SRBCT	Mean	**1**	0.96471	**1**	0.94706	**1**	0.994118	0.9294118	**1**	0.98235	**1**	0.97059	**1**	0.91176
	Std	**0**	0.03038	**0**	0.0186	**0**	0.018602	0.0248022	**0**	0.02841	**0**	0.031	**0**	0.04159
Friedman's mean rank		11.84375	6.125	11.25	6.625	9.21875	5.78125	3.84375	5.40625	4.90625	7.375	6.21875	8.4375	3.96875
Rank		1	8	2	6	3	9	13	10	11	5	7	4	12



(a) Average of fitness values (b) Average of standard deviation

Fig. 2. The average performance concerning fitness values of each optimizer across all datasets.



(a) Average of accuracy (b) Average of standard deviation

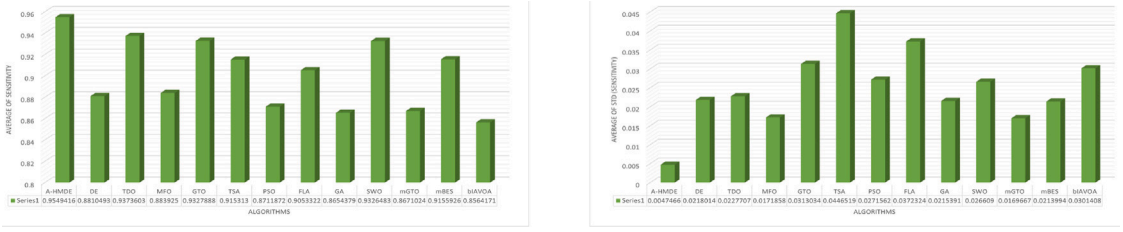
Fig. 3. The average performance of each optimizer in terms of accuracy across all datasets.

results in nine and seven datasets, respectively. Furthermore, A-HMDE achieved the highest mean rank in the Friedman test for sensitivity, demonstrating its superiority in achieving high sensitivity values. The proposed A-HMDE demonstrated superior performance in terms of average standard deviation, outperforming other methods, as depicted in Fig. 4. This observation serves as strong evidence for the methodology's stability, indicating it consistently delivered consistent results across multiple runs with minimal variance.

- Likewise, with regards to specificity, as showcased in Table 6, A-HMDE demonstrated the highest specificity value in thirteen out of sixteen datasets. Furthermore, the Friedman test revealed that A-HMDE exhibited the best mean rank in terms of specificity, hence indicating its better performance in attaining the highest

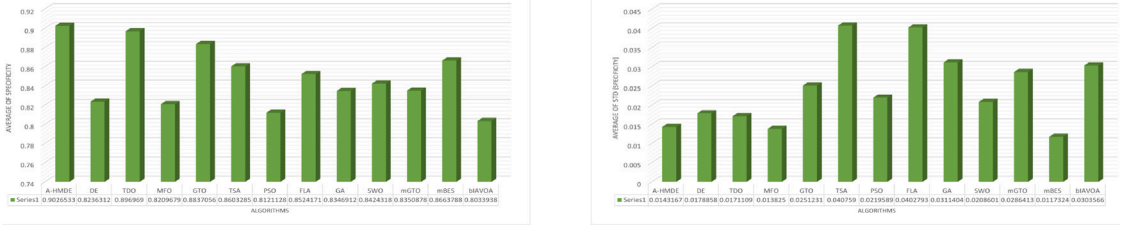
Table 5
Sensitivity comparison between A-HMDE and other optimization methods.

Dataset	Measures	A-HMDE	DE	TDO	MFO	GTO	TSA	PSO	FLA	GA	SWO	mGTO	mBES	bIAVOA
Low dimensional datasets														
Arrhythmia	Mean	**0.99861**	0.99722	0.96111	**0.99861**	0.981944	0.97361	**0.99861**	0.99722	0.99734	0.99745	0.99601	0.99468	0.98404
	Std	**0.00556**	0.00759	0.03934	**0.00556**	0.023263	0.02718	**0.00556**	0.01111	0.00727	0.00697	0.00858	0.00952	0.01822
BreastEW	Mean	**0.99907**	0.9972	**0.99907**	**0.99907**	0.996269	0.98601	0.99534	0.99067	0.9944	0.99578	0.99813	**0.99907**	0.99254
	Std	**0.00373**	0.00602	**0.00373**	**0.00373**	0.006675	0.01015	0.00714	0.01073	0.00746	0.00647	0.0051	**0.00373**	0.00771
Diabetes	Mean	0.88214	0.88274	**0.88571**	0.88274	**0.885714**	0.87619	0.86905	0.8619	0.875	0.855	0.8763	0.875	0.8776
	Std	0.01429	0.01464	3.4E−16	0.01464	3.44E−16	0.02384	0.02591	0.03052	**0**	0.01366	0.00521	**0**	0.00712
HeartEW	Mean	0.90341	0.9053	0.90341	0.9072	0.905303	0.87879	0.89773	0.89205	0.88793	**0.94907**	0.90733	0.8944	0.88793
	Std	0.01222	0.01035	0.01222	**0.00758**	0.01035	0.01565	0.01515	0.01553	0.0321	0.05388	0.03264	0.03441	0.03448
Hepatitis	Mean	**0.94118**	0.92647	**0.94118**	0.93015	0.930147	0.90809	0.89706	0.91912	0.91544	0.91176	0.93382	**0.94118**	0.91176
	Std	**1.1E−16**	0.03396	**1.1E−16**	0.02371	0.044118	0.07114	0.05882	0.04743	0.03701	0.0372	0.02009	**1.1E−16**	0.04803
Lymphography	Mean	0.97517	0.97414	0.9569	0.9569	0.953448	0.95345	0.96034	0.94134	0.98264	0.9375	**0.98611**	0.98264	**0.98611**
	Std	0.04016	0.0314	0.04458	0.0473	0.049118	0.04783	0.04227	0.04826	0.03345	0.07211	0.03208	**0.0266**	0.03208
Parkinson	Mean	**0.85714**	0.74107	0.83929	0.78571	0.830357	0.74107	0.74107	0.76786	0.625	0.78472	0.625	0.625	0.625
	Std	1.1E−16	0.05759	0.0488	0.07377	0.057588	0.05759	0.05759	0.07143	**0**	0.09488	**0**	**0**	**0**
High dimensional datasets														
Retionpathy	Mean	**0.87619**	0.82475	0.81495	0.81373	0.81924	0.82047	0.82904	0.82475	0.86571	0.825	0.86476	0.87048	0.84762
	Std	2.3E−16	0.02174	0.00607	**2.3E−16**	0.020555	0.02079	0.02177	0.02021	0.01305	0.01557	0.00984	0.0092	0.03836
Prostate	Mean	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	0.875	0.93636	0.875	0.95	0.875
	Std	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	0.04391	**0**	0.06455	**0**
Thyroid	Mean	**0.84615**	0.80769	**0.84615**	**0.84615**	0.82906	0.80769	0.79487	0.81197	0.62045	0.78235	0.67273	0.67955	0.56591
	Std	**0**	0.06013	**0**	**0**	0.041872	0.09964	0.08581	0.05296	0.11641	0.08108	0.04312	0.11002	0.04962
CNS	Mean	**1**	0.75	0.9	0.75	0.9	0.925	0.75	0.8	0.98571	**1**	0.92857	**1**	0.91429
	Std	**0**	**0**	0.1291	**0**	0.129099	0.12076	0	0.10541	0.04518	**0**	0.07529	**0**	0.07377
Colon	Mean	**1**	0.46667	0.95	0.46667	0.916667	0.83333	0.4	0.76667	0.275	**1**	0.2625	0.8375	0.3375
	Std	**0**	0.10541	0.08051	0.07027	0.087841	0.13608	0.08607	0.14055	0.0527	**0**	0.03953	0.08437	0.06038
Leukemia	Mean	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**
	Std	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**
Lung Cancer	Mean	**1**	**1**	**1**	**1**	**1**	0.99677	**1**	**1**	**1**	**1**	**1**	**1**	**1**
	Std	**0**	**0**	**0**	**0**	**0**	0.0102	**0**	**0**	**0**	**0**	**0**	**0**	**0**
Ovarian	Mean	**1**	0.82353	**1**	0.80588	0.976471	0.95882	0.80588	0.91176	0.94737	0.94737	0.94737	**1**	0.94737
	Std	**0**	0	**0**	0.02841	0.030376	0.02841	0.02841	0.04159	1.2E−16	1.2E−16	1.2E−16	**0**	1.2E−16
SRBCT	Mean	**1**	**1**	**1**	**1**	**1**	0.98571	**1**	**1**	**1**	**1**	**1**	**1**	0.95
	Std	**0**	**0**	**0**	**0**	**0**	0.04518	**0**	**0**	**0**	**0**	**0**	**0**	0.11249
Friedman's mean rank		10.4375	7	8.53125	7.75	7.875	4.84375	5.78125	5.8125	5.84375	7.09375	7.0625	7.96875	5
Rank		1	8	2	5	4	13	11	10	9	6	7	3	12



(a) Average of sensitivity (b) Average of standard deviation

Fig. 4. The average performance of each optimizer in terms of sensitivity across all datasets.



(a) Average of specificity (b) Average of standard deviation

Fig. 5. The average performance of each optimizer in terms of specificity across all datasets.

specificity values. Considering the frequent emphasis on minimizing false-positive diagnoses in medical diagnostics, achieving a high level of specificity is of utmost significance. The comparative analysis of A-HMDE with other methods, as shown in Fig. 5, provides further evidence to confirm the stability and dependability of A-HMDE, as indicated by its lower average standard deviation.

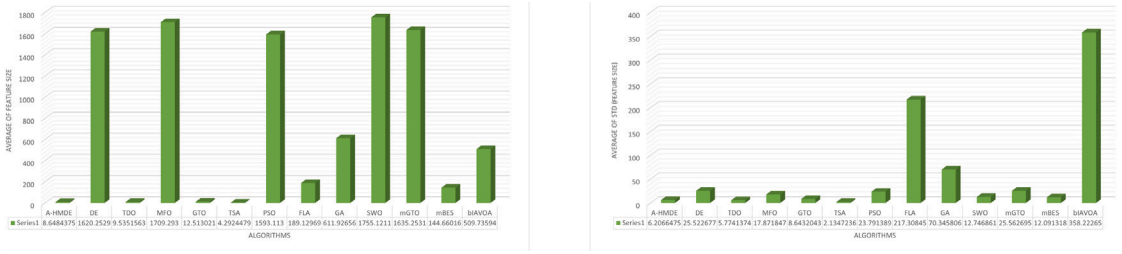
- When considering the average number of selected features, Table 7 provides a comparison between the number of features chosen by the A-HMDE method and its competitors. After a comprehensive analysis of the results, it becomes clear that A-HMDE surpasses other methods by achieving the highest average number of selected features in seven out of the sixteen datasets utilized in this study. Meanwhile, the TSA method outperformed other methods in six datasets. The highest Friedman mean rank of feature selection size was achieved by TSA, while A-HMDE ranked second. These results highlight the proficiency of the A-HMDE algorithm in identifying relevant features with a high degree of

Table 6
Specificity comparison between A-HMDE and other optimization methods.

Dataset	Measures	A-HMDE	DE	TDO	MFO	GTO	TSA	PSO	FLA	GA	SWO	mGTO	mBES	bIAVOA
Low dimensional datasets														
Arrhythmia	Mean	**0.52446**	0.36005	0.52174	0.35326	0.415761	0.37364	0.33424	0.32609	0.38352	0.20982	0.38778	0.52131	0.30824
	Std	0.08434	0.04488	0.08017	0.05175	0.068688	0.06746	0.06588	0.09392	0.04685	0.03711	0.04581	**0.03368**	0.04618
BreastEW	Mean	**0.96144**	0.95878	0.96011	0.95878	0.945479	0.94016	0.94548	0.94415	0.9508	0.94844	0.95878	0.95878	0.9508
	Std	**0.00777**	0.00942	0.01064	0.00942	0.013386	0.01394	0.01548	0.01715	0.01019	0.01106	0.00942	0.00942	0.01019
Diabetes	Mean	0.59694	0.59311	0.59184	0.59311	0.591837	0.59949	0.6148	0.6199	0.58621	**0.67361**	0.58405	0.58621	0.5819
	Std	0.02041	0.02631	2.3E−16	0.02631	2.29E−16	0.04207	0.03641	0.05263	**0**	0.03163	0.00862	**0**	0.01178
HeartEW	Mean	0.95536	0.95238	**0.96131**	0.95536	0.958333	0.94048	0.95238	0.9494	0.855	0.84491	0.84	0.86	0.855
	Std	0.0119	**1.1E−16**	0.0192	0.0119	0.016265	0.02749	0.01739	0.0119	0.05441	0.051	0.05657	0.06367	0.0459
Hepatitis	Mean	0.6875	0.67857	0.66071	0.66518	0.660714	0.66518	0.63839	0.625	0.60268	**0.69643**	0.6875	0.65179	0.66071
	Std	0.04518	0.0777	0.03194	0.06237	0.061168	0.08131	0.06634	0.04124	0.05814	0.05533	0.06322	**0.0244**	0.07143
Lymphography	Mean	**1**	0.99784	0.99138	0.99569	0.991379	0.98491	0.98707	0.98276	0.79688	0.92434	0.79167	0.80208	0.77083
	Std	**0**	0.00862	0.02356	0.01724	0.015421	0.03554	0.02479	0.02181	0.06783	0.05425	0.06086	0.05159	0.03727
Parkinson	Mean	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	0.96774	0.96458	0.96774	0.96774	0.96774
	Std	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	4.6E−16	0.03096	4.6E−16	4.6E−16	4.6E−16
High dimensional datasets														
Retionpathy	Mean	**0.74176**	0.72093	0.73692	0.73983	0.719477	0.68556	0.70252	0.69138	0.71111	0.68976	0.71905	0.71667	0.71032
	Std	0.00677	0.03165	0.01077	0.01058	0.027476	0.016	0.02981	0.0216	0.01071	0.01826	0.0041	**0.00383**	0.03331
Prostate	Mean	**1**	0.8125	**1**	0.8	**1**	0.975	0.8	0.9	0.91538	0.98	0.93846	**1**	0.83077
	Std	**0**	0.03536	**0**	1.2E−16	**0**	0.04629	1.2E−16	0.09258	0.04367	0.04216	0.03243	**0**	0.03243
Thyroid	Mean	**1**	0.99964	**1**	**1**	0.999881	0.99905	0.99964	0.99964	0.99635	0.99701	0.9985	0.99749	0.99542
	Std	**0**	0.0006	**0**	**0**	0.000291	0.00133	0.00039	0.00039	0.00153	0.00201	0.00023	0.00113	0.00255
CNS	Mean	**0.975**	0.75	0.9375	0.75	0.925	0.7875	0.725	0.8375	0.8	0.8	0.8	0.8	0.74
	Std	0.0527	**0**	0.06588	**0**	0.087401	0.11859	0.0527	0.10291	1.2E−16	1.2E−16	1.2E−16	1.2E−16	0.09661
Colon	Mean	**1**	**1**	**1**	**1**	**1**	0.98571	**1**	**1**	**1**	**1**	**1**	**1**	**1**
	Std	**0**	**0**	**0**	**0**	**0**	0.04518	**0**	**0**	**0**	**0**	**0**	**0**	**0**
Leukemia	Mean	**1**	0.71429	**1**	0.71429	0.971429	0.92857	0.71429	0.84286	0.925	0.75	0.9	**1**	0.75
	Std	**0**	**0**	**0**	**0**	0.060234	0.07529	**0**	0.12509	0.12076	**0**	0.1291	**0**	**0**
Lung Cancer	Mean	**1**	0.7	0.99	0.7	0.96	0.9	0.7	0.92	0.89167	**1**	0.83333	**1**	0.84167
	Std	**0**	1.2E−16	0.03162	1.2E−16	0.05164	0.08165	1.2E−16	0.06325	0.04025	**0**	1.2E−16	**0**	0.02635
Ovarian	Mean	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**	**1**
	Std	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**	**0**
SRBCT	Mean	**1**	0.94	**1**	0.91	**1**	**1**	0.88	**1**	0.97273	**1**	0.95455	**1**	0.89091
	Std	**0**	0.05164	**0**	0.03162	**0**	**0**	0.04216	**0**	0.04391	**0**	0.04791	**0**	0.07171
Friedman's mean rank		11.0938	6.8125	10.0313	7.15625	8.84375	6.4375	5.0625	6.625	5.3125	6.40625	5.96875	7.40625	3.84375
Rank		1	6	2	5	3	8	12	7	11	9	10	4	13

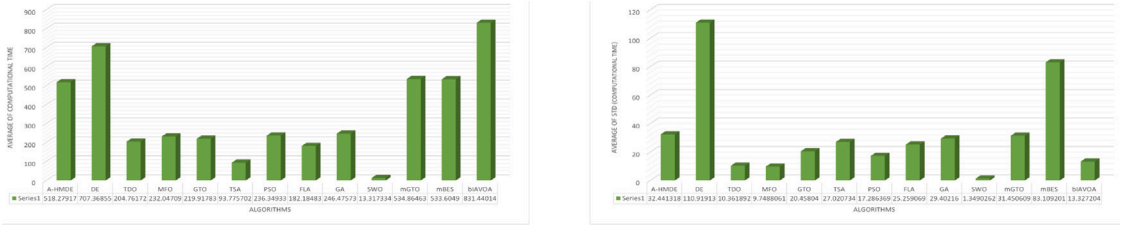
Table 7
No of features comparison between A-HMDE and other optimization methods.

Dataset	Measures	A-HMDE	DE	TDO	MFO	GTO	TSA	PSO	FLA	GA	SWO	mGTO	mBES	bIAVOA
Low dimensional datasets														
Arrhythmia	Mean	40	130.25	17.5625	128.063	19.3125	**7.625**	106.3125	127.625	88.75	140	128.875	28	73.8125
	Std	36.90528	8.29859	9.99979	7.576	11.98176	**5.27731**	8.6773172	44.1511	10.2209	7.56307	7.34734	12.6122	50.3656
BreastEW	Mean	7.5	9.125	8.9375	10.375	10.375	**6.3125**	10	11.625	11	14.625	10.875	8.125	17.75
	Std	1.460593	1.78419	1.98221	1.5864	2.362908	1.8518	2.2803509	3.66742	2.47656	2.62996	**1.31022**	1.40831	2.26569
Diabetes	Mean	2.0625	2.0625	**2**	2.0625	**2**	2.1875	2.4375	2.5	3	5.375	3.0625	3	3.125
	Std	0.25	0.25	**0**	0.25	**0**	0.40311	0.813941	0.63246	0	0.5	0.25	0	0.34157
HeartEW	Mean	6.375	6.8125	6.4375	7.1875	7.8125	**4.0625**	7.1875	8.1875	6.25	6.625	6.375	6.1875	7.1875
	Std	1.78419	1.97379	0.72744	0.75	1.276388	1.48183	1.5585784	3.41016	1.06458	1.31022	**0.5**	0.75	0.65511
Hepatitis	Mean	2.6875	5.4375	2.25	4.5625	2.25	**1.9375**	4	3.125	5.0625	8.125	4.5	2.1875	6.625
	Std	1.25	2.47572	0.44721	2.12818	0.68313	0.57373	1.7511901	2.15639	2.20511	2.15639	2.3094	**0.40311**	3.05232
Lymphography	Mean	6	6.6875	5.25	6.0625	6.0625	**4.3125**	7.0625	9	6.5	9.125	7.0625	5.0625	10.4375
	Std	2.160247	1.81544	1.91485	2.29401	2.434988	**1.01448**	1.8427787	1.93218	2.70801	1.5	1.76895	2.76812	2.39357
Parkinson	Mean	2	2	1.9375	2	1.875	**1.375**	3.3125	1.9375	2.5625	8.0625	2	2	5.4375
	Std	**0**	0.36515	0.44253	0.63246	0.5	0.5	1.3022417	0.7719	0.62915	1.65202	**0**	**0**	1.59034
High dimensional datasets														
Retionpathy	Mean	5	7.0625	6.6875	7	6.5625	4.875	7.4375	7.125	5.2	8.4	**3.9**	4.4	8.5
	Std	**0**	1.38894	0.70415	0.8165	1.631717	1.25831	1.1528949	2.24722	1.13529	2.17051	0.99443	0.96609	1.95789
Prostate	Mean	**2.25**	4879.38	5	5096.38	24.125	3.625	4751.625	1596.75	2363.3	5219.7	5027.2	5.8	2264.3
	Std	**0.46291**	105.16	4.30946	46.5708	23.673	2.19984	78.845124	1783.61	270.194	67.4389	104.18	7.88529	2146.05
Thyroid	Mean	**4**	4.33333	**4**	**4**	4.333333	5.16667	5.3333333	4.5	5.8	7.3	4.9	4.8	9.3
	Std	**0**	0.5164	**0**	**0**	0.516398	0.75277	1.0327956	1.3784	1.39841	1.63639	0.31623	0.78881	1.56702
CNS	Mean	48.9	3206.6	49.8	3383	33.2	**5.7**	3155	606.9	1340.6	3503.1	3291.5	2223.9	1426.5
	Std	52.18652	37.0141	35.1119	30.5869	14.45145	**2.26323**	59.537103	517.466	74.5717	31.3527	50.1093	152.708	1126.04
Colon	Mean	**3.3**	806.6	4.9	898.5	5.1	**3.3**	767.6	36.1	221.5	946.4	746.2	5.1	235.7
	Std	**1.05935**	32.291	4.06749	29.1519	3.414023	1.33749	21.624061	48.6106	24.158	7.60409	29.5439	4.09471	119.715
Leukemia	Mean	**1.5**	3134	5.5	3368	8.4	3.6	3140.4	181.4	1279.8	3501.2	3215.3	2.1	1399.6
	Std	**0.527046**	29.6086	6.5532	33.7935	5.146736	3.47051	31.355311	454.79	218.688	18.7249	65.8804	0.8756	1140.13
Lung Cancer	Mean	**3**	5610.8	25.5	6013.8	50.9	6	5647.9	142.4	2112.5	6174.5	5804.5	8.7	1132.2
	Std	**0.471405**	96.6549	21.0357	21.4258	53.21539	5.84998	60.916062	196.406	326.347	17.4372	60.4635	6.16532	436.257
Ovarian	Mean	**2.8**	7116.8	4.3	7373.1	11.7	4.2	6938.4	224.5	1841.8	7419.8	6897.9	2.9	1027.1
	Std	**0.788811**	59.7435	2.54078	87.1467	12.01897	3.9101	80.786688	317.232	142.652	27.8799	47.0353	0.8756	273.155
SRBCT	Mean	**1**	996.1	2.5	1044.6	6.2	4.4	935.8	62.4	497.2	1109.6	1009.9	2.3	528.2
	Std	**0**	29.0228	2.54951	21.2404	4.98442	2.01108	27.185781	98.4696	47.0834	12.3935	36.9938	1.1595	426.029
Friedman's mean rank		2.9375	8.65625	3.59375	9.1875	4.8125	2.53125	9.15625	7.40625	7.71875	12.375	8.875	4	9.75
Rank		2	8	3	11	5	1	10	6	7	13	9	4	12



(a) Average of feature size (b) Average of standard deviation

Fig. 6. The average performance of each optimizer in terms of feature size across all datasets.



(a) Average of computational times (b) Average of standard deviation

Fig. 7. The average performance of each optimizer in terms of computational times across all datasets.

accuracy across various datasets. By inspecting standard deviation, as shown in Fig. 6, relative to other techniques, A-HMDE is a robust method for most of the datasets.

- In terms of average computational time, which signifies an algorithm's speed in selecting features from a given dataset, this study presents a comparison in Table 8 between the mean computational time results of the A-HMDE approach and alternative optimization methods. Across most experimental scenarios involving various optimization algorithms, the A-HMDE approach consistently showcased superior performance, except concerning time consumption, as outlined in Table 8. The findings underscore the A-HMDE's ability to strike a nuanced equilibrium between exploration and exploitation. Conversely, TSA exhibited superior time consumption outcomes across all datasets due to its vulnerability to premature convergence. This phenomenon leads to algorithms getting trapped in local optima, thereby hindering them from finding the best candidate solution. Consequently, TSA emerged as the fastest optimizer, as indicated by Friedman's mean rank for consumption time. Moreover, the results, based on Friedman's mean rank, position A-HMDE in the eighth spot. Despite A-HMDE's marginally higher time consumption, as visually demonstrated in Fig. 7, its adept balance between exploration and exploitation substantially contributed to its robust and reliable performance.

5.1.4. Graphical analysis

This subsection offers a graphical analysis of the convergence curves for the compared algorithms. Additionally, it includes an examination of the boxplot illustrating the performance of the A-HMDE method in comparison to the other evaluated algorithms.

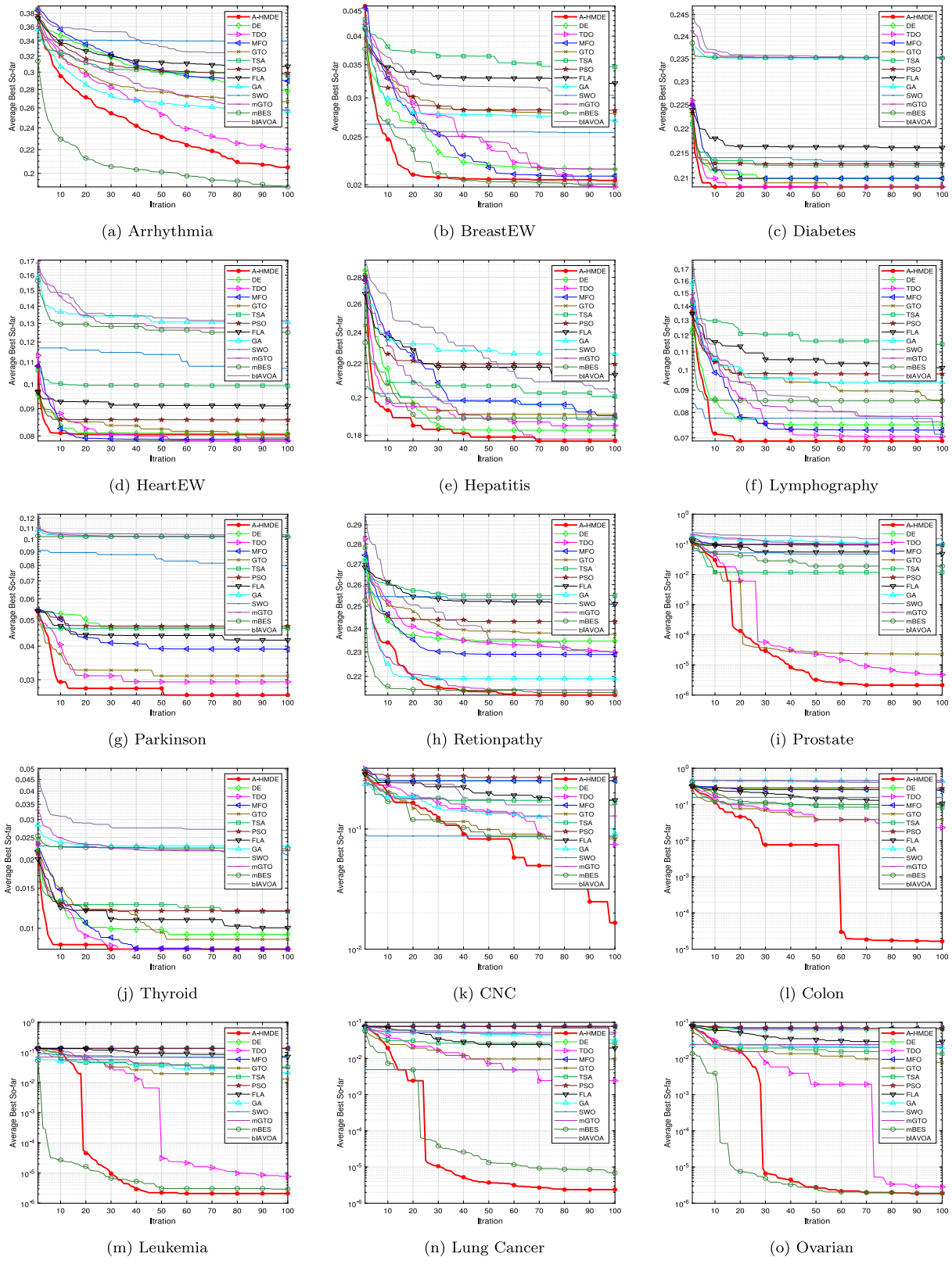
The convergence curve represents the relationship between the optimal solutions obtained by the algorithm and the number of iterations, which indicates the algorithm's ability to maximize or minimize the fitness function and the speed of reaching the solution. Fig. 8 compares the convergence curves of A-HMDE and competing algorithms to demonstrate the differences in reaching the near-optimal solution in a smaller number of iterations. These curves serve as evidence that A-HMDE efficiently identified optimal solutions within a shorter timeframe compared to its competitors across diverse datasets. Specifically, A-HMDE demonstrated the fastest convergence towards the optimal solution in eighteen out of twenty datasets, indicating its superior performance in solving the given problems.

A boxplot serves as a graphical portrayal of statistical information, summarizing numerous numerical variables through their quartiles. Employing boxplot analysis facilitates the representation of data distribution attributes. These graphs effectively categorize data distribution into three core quartiles: upper, lower, and middle, while the whiskers signify the outermost data points that the algorithm reaches — the minimum and maximum. The borders of the boxes delineate the lower and upper quartiles. A narrow boxplot signifies strong agreement among data points. Illustrated in Fig. 9 is a boxplot portraying the outcomes across 15 distinct disease datasets, each characterized by varying feature sizes. For all the disease datasets, the boxplots generated by the A-HMDE algorithm are notably narrower compared to those of the alternative algorithms, indicating lower values. Indeed, across all disease datasets, the A-HMDE algorithm exhibited superior performance in comparison to other algorithms.

5.1.5. Comparison between different variants of DE algorithm and A-HMDE

The A-HMDE framework incorporates the integration of three distinct strategies into the fundamental DE algorithms. To demonstrate the efficacy of this combination, two variants of DE are developed and evaluated in comparison to A-HMDE. Table 9 provides a detailed overview of these DE variants. The symbol "N" signifies that this version does not incorporate the current mechanism, while "Y" indicates that this variant includes the current mechanism. Table 10 displays the averages for various metrics, including fitness values, classification accuracy, sensitivity, specificity, feature size, and computational time. These values are computed based on 30 independent runs across different datasets utilized in this study.

Regarding average fitness values, A-HMDE surpassed the DE1 algorithm in fifteen datasets, and it outperformed DE2 in thirteen datasets. On the other hand, both DE1 and DE2 outperformed the original DE across all datasets, underscoring the impact of integrating the two strategies within one method. This observation is further reinforced by the classification accuracies presented in the same table, where the proposed A-HMDE method outperformed the DE1 algorithm in thirteen datasets and surpassed DE2 in ten datasets. A-HMDE's superiority in FS is evident from its outperformance of DE1 in fifteen datasets and DE2 in twelve datasets, indicating its ability to select the most informative



(a) Arrhythmia (b) BreastEW (c) Diabetes

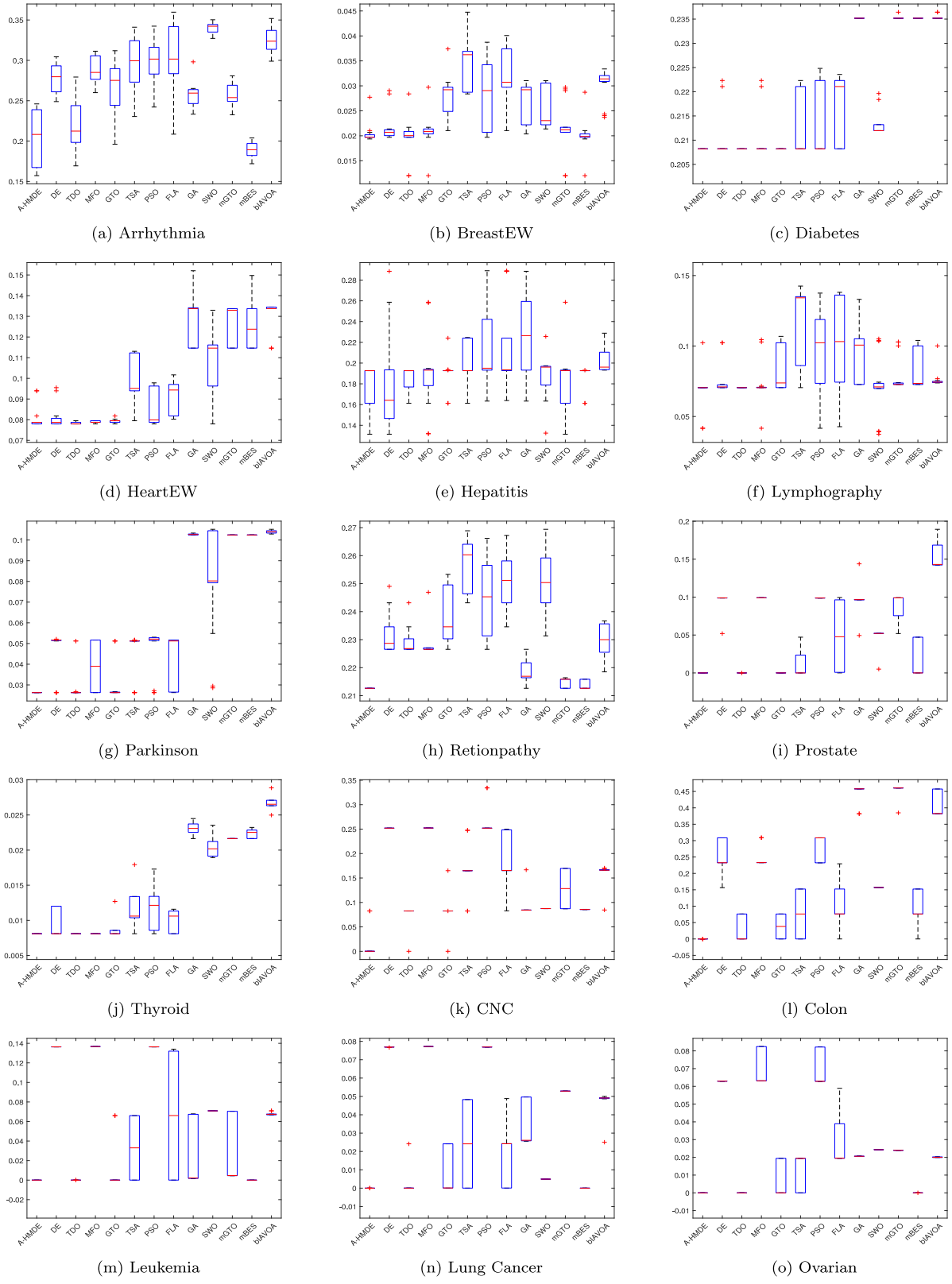
(d) HeartEW (e) Hepatitis (f) Lymphography

(g) Parkinson (h) Retionpathy (i) Prostate

(j) Thyroid (k) CNC (l) Colon

(m) Leukemia (n) Lung Cancer (o) Ovarian

Fig. 8. Illustration of the average convergence speed across algorithms on various datasets.



(a) Arrhythmia (b) BreastEW (c) Diabetes

(d) HeartEW (e) Hepatitis (f) Lymphography

(g) Parkinson (h) Retionpathy (i) Prostate

(j) Thyroid (k) CNC (l) Colon

(m) Leukemia (n) Lung Cancer (o) Ovarian

Fig. 9. Boxplots comparing A-HMDE with other optimization methods for fitness values.

Table 8
Computational times comparison between A-HMDE and other optimization methods.

Dataset	Measures	A-HMDE	DE	TDO	MFO	GTO	TSA	PSO	FLA	GA	SWO	mGTO	mBES	bIAVOA
Low dimensional datasets														
Arrhythmia	Mean	165.325	418.612	30.9453	51.1809	38.19309	10.7277	52.5087	60.3697	76.8317	3.75585	155.632	171.824	255.259
	Std	83.0887	1057.54	13.7839	21.9329	23.27887	6.32236	29.0533	31.0665	2.4219	2.44045	4.09504	21.4592	11.4944
BreastEW	Mean	75.7682	52.7096	19.3198	14.5303	26.12747	9.67061	16.0645	21.3598	29.3283	1.93249	60.8511	86.8472	92.8368
	Std	39.3174	36.1994	10.9717	8.73917	17.64343	7.22945	10.6983	13.3474	15.939	1.12154	29.7497	44.154	42.5798
Diabetes	Mean	56.7591	57.2075	26.8622	18.2366	29.09994	12.122	18.7387	18.4761	35.1756	1.41601	72.7456	145.401	79.824
	Std	8.38245	9.44343	4.37865	2.97633	4.019917	4.90754	3.02256	2.73371	16.3561	0.99928	29.2451	151.968	38.0578
HeartEW	Mean	29.2867	11.8143	6.60726	4.47711	7.765383	3.2874	4.88781	4.96512	6.78963	0.26546	12.9727	31.9258	14.1689
	Std	11.5946	4.34159	3.69225	2.44864	4.967006	1.89846	3.21471	3.01316	1.14828	0.0387	2.58375	4.18927	2.33986
Hepatitis	Mean	47.5973	21.1063	9.69123	6.63346	9.786355	3.56205	7.19816	6.9643	9.20867	0.17133	18.2921	48.1294	18.3868
	Std	15.5852	7.11172	3.57574	2.59452	3.56895	1.74913	2.45089	2.09699	3.02906	0.03244	4.82786	9.66763	5.89207
Lymphography	Mean	39.9282	17.3416	8.54482	5.87482	8.922918	3.62645	5.33682	5.60256	6.00303	0.18378	12.9391	40.2436	12.7183
	Std	19.0628	8.89264	4.31348	3.08581	4.460192	1.74582	2.87018	2.64102	2.1473	0.03085	5.3402	7.78626	4.32783
Parkinson	Mean	27.9137	12.51	6.51456	4.32646	6.519764	1.79051	4.10587	3.96392	5.17956	0.19216	10.3052	34.198	12.6387
	Std	18.6879	8.71319	4.64987	3.02044	4.639246	1.24479	2.97535	2.87895	0.81225	0.02978	0.42155	1.68912	0.49706
High dimensional datasets														
Retionpathy	Mean	53.9866	218.955	105.557	69.6424	108.7854	44.9947	68.1279	71.3491	52.4394	5.18501	115.441	155.743	213.436
	Std	92.0944	106.892	49.9295	35.0041	55.52454	23.3081	34.9929	32.4661	2.77071	0.24476	5.00959	8.62089	3.57085
Prostate	Mean	223.301	337.378	27.2544	118.382	27.81109	12.6338	112.629	84.8468	105.136	11.6514	246.038	40.4471	358.234
	Std	23.4841	3.96261	3.32489	1.11077	5.25434	1.23789	1.36165	33.9698	3.72833	0.5928	4.02935	6.2965	1.91168
Thyroid	Mean	6479.86	5917.01	2856.95	1946.14	3022.735	1326.47	2103.18	2135.95	2627.26	97.9213	4832.79	7263.16	6723.26
	Std	120.645	381.646	52.4083	39.3737	167.483	377.94	136.452	150.944	385.362	10.6293	346.218	1034.48	30.4185
CNS	Mean	148.562	106.44	17.9624	38.8422	16.89601	8.53117	35.161	24.5995	34.7208	2.95594	88.5619	172.588	120.343
	Std	10.6736	1.39232	0.89209	0.89796	1.746876	0.30758	0.60517	4.58028	1.93968	0.13132	2.95716	1.64624	3.02985
Colon	Mean	32.8979	43.6714	6.30881	15.4764	7.415207	3.99756	14.9925	9.00804	13.9593	1.25884	33.5654	14.225	51.3184
	Std	1.53991	0.95528	0.25174	0.17625	0.354846	0.65583	0.36721	1.2416	0.38946	0.15326	1.73911	1.08402	0.67665
Leukemia	Mean	127.304	150.803	14.6767	54.0201	17.33162	8.28661	50.5474	25.4366	45.3864	8.71429	117.764	25.0182	163.681
	Std	2.64067	3.53892	1.5148	1.45319	1.619719	0.62648	1.32097	5.92832	2.9962	0.63744	1.8663	2.94813	3.8084
Lung Cancer	Mean	288.425	1298.19	56.4887	460.235	74.29539	20.2853	431.891	139.048	334.586	34.4835	931.012	130.157	1721.95
	Std	13.9717	26.5868	3.69835	5.93852	9.741522	0.87972	8.06148	30.4048	15.0045	2.44459	14.6403	15.0515	28.4627
Ovarian	Mean	435.864	2587.24	73.2455	880.564	107.2079	29.1398	834.329	288.498	539.504	41.003	1794.65	159.595	3380.46
	Std	56.0526	115.56	7.4835	26.7393	21.76306	1.71436	38.5177	84.3729	15.3973	1.96428	48.5068	17.1472	33.2431
SRBCT	Mean	59.6921	66.904	9.25659	24.195	9.792633	1.28139	21.888	14.5222	22.106	1.98695	54.2716	18.1794	84.5306
	Std	2.23984	1.93416	0.92144	0.48914	1.263111	0.56452	0.61776	2.4595	0.99241	0.09364	1.97958	1.5604	2.92473
Friedman's mean rank		10.0625	11.125	4.875	6.25	5.875	1.75	6	5.25	6.75	1.25	10	9.875	11.9375
Rank		11	12	3	7	5	2	6	4	8	1	10	9	13

Table 9
Different DE variants using three different strategies.

Algorithms	Adaptive parameter	Hybrid mutation strategies	ESQ
DE	N	N	N
DE1	Y	N	Y
DE2	Y	Y	N
A-HMDE	Y	Y	Y

features, thereby contributing to higher accuracy rates. Additionally, concerning average computational time, A-HMDE outperformed the DE1 algorithm in eleven datasets and surpassed DE2 in nine datasets, underscoring its efficiency in terms of processing speed.

Based on the experimental outcomes, all three DE variants demonstrate notable performance improvements compared to the standard DE, with A-HMDE emerging as the most powerful variation. In simpler terms, A-HMDE outperforms the others significantly. Additionally, Table 11 presents the mean values of the Friedman test results across 16 datasets, showcasing A-HMDE's superior performance compared to other variants. This is further reinforced by Fig. 10, a visual representation of the results, clearly indicating A-HMDE's dominance over the alternative variants.

5.2. Experiment 2: Skin diseases prediction

In the second phase of experimentation, the efficacy of the proposed A-HMDE method is extended through its application to a practical scenario involving the identification of skin diseases. The experimental framework for this application is organized into four key steps, as illustrated in Fig. 11:

- Preprocessing: To ensure consistency and compatibility, all input images are resized to a uniform size of 224 × 224 pixels. Additionally, the images are transformed into the RGB color space with an 8-bit depth.
- Feature extraction: In this step, two distinct pre-trained convolutional neural network (CNN) models, namely Vgg16 [71] and Resnet50 [72], are utilized for deep feature extraction. These models have been pre-trained on large-scale datasets and have demonstrated strong capabilities in extracting informative features from images.
- Feature selection: The proposed A-HMDE method is applied as a FS algorithm to identify the most relevant and distinctive features from the concatenated deep features obtained in the previous step. By leveraging the optimization capabilities of A-HMDE, the algorithm aims to select a subset of features that maximizes the discriminative power and minimizes redundancy.
- Classification: In the final step, a KNN classifier is employed to classify the selected features. The KNN classifier is a well-known and widely used algorithm for pattern recognition tasks. By utilizing the selected features, the KNN classifier is able to make accurate predictions and classify skin diseases effectively.

For the validation of the proposed framework, a freely available dataset called DermNet is utilized. This dataset comprises approximately 23,000 images that have been collected and labeled by the Dermnet Skin Disease Atlas. From this dataset, sample images representing various skin diseases such as acne, keratosis, Eczema herpeticum, and melanoma are selected. These specific diseases are chosen to evaluate the effectiveness and performance of the proposed framework in accurately identifying

Table 10
The average values of several performance metrics for different DE variations.

Dataset	Average fitness values				Classification accuracy				Sensitivity				Specificity			
	A-HMDE	DE1	DE2	DE	A-HMDE	DE1	DE2	DE	A-HMDE	DE1	DE2	DE	A-HMDE	DE1	DE2	DE
Low dimensional datasets																
Arrhythmia	**0.204909**	0.256509	0.246666	0.278222	**0.794469**	0.745052	0.755184	0.723684	0.998611	**0.99734**	**1**	0.997222	**0.524457**	0.389205	0.440217	0.360054
BreastEW	0.020411	0.020077	**0.019868**	0.021496	0.981908	**0.982456**	**0.982456**	0.98136	0.999067	0.999067	**1**	0.997201	**0.961436**	0.958777	0.948718	0.958777
Diabetes	0.209096	**0.208214**	0.216719	0.2099	0.791396	**0.792208**	0.784091	0.790584	0.882143	**0.885714**	0.866667	0.882738	0.596939	0.591837	**0.607143**	0.593112
HeartEW	**0.080529**	0.087548	0.102532	0.080865	**0.923611**	0.916667	0.900463	**0.923611**	0.903409	0.890152	**0.918561**	0.905303	**0.955357**	0.958333	0.872024	0.952381
Hepatitis	0.17706	0.17173	**0.168165**	0.182499	0.822581	0.828629	**0.832661**	0.818548	0.941176	0.9375	**0.947368**	0.926471	0.6875	**0.696429**	0.651042	0.678571
Lymphography	**0.068867**	0.074467	0.091395	0.074978	**0.933804**	0.929095	0.911925	0.928017	0.975172	0.993056	0.955172	0.974138	**1**	0.796875	0.997845	0.997845
Parkinson	**0.026294**	0.027231	0.026748	0.046919	**0.974359**	**0.974359**	**0.974359**	0.953526	0.857143	0.876535	**0.8875**	0.741071	**1**	0.967742	0.987132	**1**
Retionpathy	**0.212632**	0.214624	0.233692	0.23461	**0.787879**	0.785281	0.766234	0.766775	**0.87619**	0.864762	0.853398	0.824755	**0.741764**	0.719048	0.696094	0.72093
Prostate	**2.14E−06**	0.018863	0.015717	0.093036	**1**	0.980952	0.984127	0.910714	**1**	0.95	**1**	**1**	**1**	**1**	**1**	0.8125
Thyroid	0.008097	0.008097	**2.65E−06**	0.009402	0.993746	0.993746	**1**	0.992587	0.846154	0.846154	0.8	0.807692	**1**	**1**	0.999288	0.999643
CNS	**0.016578**	0.123929	0.07432	0.251998	**0.983333**	0.875	0.925	0.75	**1**	0.85	0.9	0.75	**0.975**	0.8875	0.9375	0.75
Colon	**1.65E−05**	0.091468	0.39829	0.247725	**1**	0.907692	0.6	0.753846	**1**	0.833333	0.35	0.466667	**1**	0.971429	**1**	**1**
Leukemia	**2.1E−06**	0.070911	0.002824	0.136396	**1**	0.933333	**1**	0.866667	**1**	1	**1**	**1**	**1**	0.75	**1**	0.714286
Lung Cancer	**2.38E−06**	0.052899	0.046213	0.076892	**1**	0.95122	0.956098	0.926829	**1**	1	**1**	**1**	**1**	0.833333	0.85	0.7
Ovarian	**1.85E−06**	0.024308	0.021949	0.062932	**1**	0.980392	0.980392	0.941176	**1**	0.947368	0.947368	0.823529	**1**	**1**	**1**	**1**
SRBCT	**4.33E−06**	0.019625	0.002993	0.039257	**1**	0.982353	**1**	0.964706	**1**	1	**1**	**1**	**1**	0.972727	**1**	0.94

Feature Size				Time			
A-HMDE	DE1	DE2	DE	A-HMDE	DE1	DE2	DE
40	114.6875	119.9375	130.25	**165.3246**	273.0803	484.3196	418.6117
7.5	8.125	**7.5**	9.125	75.76821	86.84715	84.17888	**52.7096**
2.0625	**2**	2.375	2.0625	56.75914	350.641	**17.05155**	57.20751
6.375	6.5625	**5.1875**	6.8125	29.28672	**4.185758**	23.95845	11.8143
2.6875	3.9375	4.75	5.4375	47.59727	24.49128	**11.5788**	21.10633
6	7.6875	7.5625	6.6875	39.92818	**7.147331**	16.53477	17.34162
2	4.0625	3	**2**	27.91374	28.45971	33.81951	**12.50999**
5	3.9	**4.3**	7.0625	**53.98665**	115.4414	61.39176	218.9547
2.25	5.8	2.833333	4879.375	223.3008	240.4471	**218.8675**	337.3781
4	4	**1.583333**	4.333333	6479.857	6479.857	**3143.419**	5917.013
48.9	127.8	49.8	3206.6	148.5622	26.52618	**17.96235**	106.4398
3.3	16.7	458	806.6	**32.8979**	85.84596	84.13766	43.67139
1.5	501.2	213	3134	**127.304**	162.7143	542.2077	150.8027
3	54.5	64.6	5610.8	**288.4247**	931.0124	293.6989	1298.195
2.8	419.8	345.6	7116.8	435.8638	541.003	991.1302	2587.238
1	47.2	69.8	996.1	**59.69207**	122.106	112.6927	66.90399

Table 11
The average Friedman test results for various performance measures.

Measures	A-HMDE	DE1	DE2	DE
Fitness Values	1.4062	2.5312	2.375	3.6875
Accuracy	3.4375	2.59375	2.5625	1.40625
Sensitivity	2.96875	2.5625	2.625	1.84375
Specificity	3.46875	2.21875	2.375	1.9375
Feature Size	1.4375	2.53125	2.40625	3.625
Time	2.53125	2.03125	2.625	2.8125

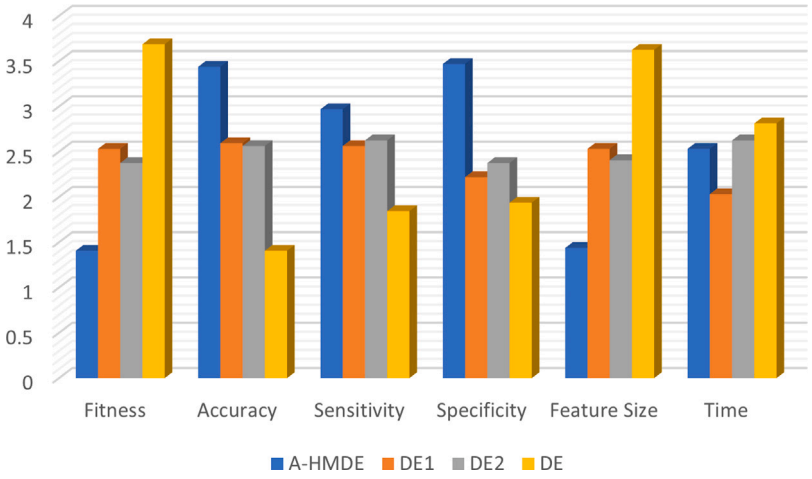


Fig. 10. The Friedman ranking of different DE variants.

and classifying different skin conditions. By utilizing these sample images from the DermNet dataset, the proposed framework can be thoroughly validated, and its capability in handling real-world skin disease detection tasks can be assessed.

In the context of skin disease prediction experiments, a comprehensive evaluation of the proposed A-HMDE method and competing methods is presented in Table 12, encompassing fitness, accuracy, sensitivity, specificity, and feature size. The results depicted in this table clearly illustrate the superiority of the proposed method in terms of fitness values. Furthermore, the small standard deviation values indicate consistent and reliable outcomes across multiple runs of the same case, underscoring the method's effectiveness in addressing real-world problems through the selection of highly relevant features.

Analyzing the classification accuracy, as presented in Table 12, serves to demonstrate the effective performance of the proposed method, consistently aligning with observed fitness function values. These high accuracy rates substantiate the proficiency of the proposed method in accurately predicting skin diseases. Additionally, the sensitivity analysis reveals that A-HMDE, PSO, and MFO achieved the highest sensitivity value of 1, with DE following closely at 0.997. In terms of specificity, A-HMDE achieved a notable value of 0.8101. Given the crucial need to minimize false-positive diagnoses in medical diagnostics, a high specificity value is essential. Furthermore, the number of selected features, as indicated in Table 12, highlights the efficacy of the proposed method in achieving an optimal feature subset while maintaining a high accuracy rate.

Fig. 12(a) illustrates the convergence curve and boxplot, providing insights into the performance of the proposed A-HMDE compared to other competitors using the skin diseases dataset. Analyzing the convergence curve, it becomes evident that the A-HMDE algorithm significantly enhances the convergence speed, enabling the rapid attainment of optimal solutions for the given problems. Furthermore, the convergence curves of the proposed A-HMDE method exhibit a close correspondence between the ideal fitness values and the optimal accuracy rates, indicating the effectiveness of the algorithm in achieving both fitness optimization and accurate classification.

Additionally, Fig. 12(b) presents the boxplot, which offers a visual representation of the distribution of obtained results and their stability across multiple runs for the given problem. When the results exhibit a small range of values, it confirms the robustness and stability of the tested method in consistently solving the problem with similar performance. In this regard, the boxplot of the proposed method demonstrates a high degree of similarity among the results, indicating the algorithm's stability in addressing the skin diseases dataset. Moreover, the low distribution values of the obtained results further support the effectiveness of the proposed method, while other competitors exhibit higher distribution values, indicating a less consistent performance.

6. Conclusions and future works

This paper introduced an adaptive hybrid-mutated DE (A-HMDE) method that offers a robust solution to the challenges posed by feature selection (FS) in classification processes, particularly in scenarios of high-dimensional datasets with limited samples. The A-HMDE incorporates four innovative strategies: (1) Hybrid Mutation Strategies, integrating SWO algorithm mechanisms to enhance performance and achieve accurate and fast convergence to global optima. (2) Adaptive Parameter Strategy, optimizing key DE parameters for more effective search. (3) Adaptive Mutation Operator, ensuring a balanced exploration–exploitation trade-off during optimization. (4) Enhanced Solution Quality (ESQ) concept, aiding DE in escaping local optima and improving the accuracy of obtained solutions. The effectiveness of the A-HMDE approach is assessed by employing it as a FS technique on datasets sourced from the UCI repository, microarray data, and skin disease image dataset. The experimental results distinctly highlight the method's adeptness in avoiding local minima and accelerating the convergence process. A thorough comparison with state-of-the-art algorithms further accentuates the remarkable accomplishments of the proposed method, with achieved accuracy levels spanning from 0.88 to 1.00. In addition to the aforementioned insights, it is imperative to highlight that the empirical evidence unequivocally supports the proposed A-HMDE method's superiority over comparative

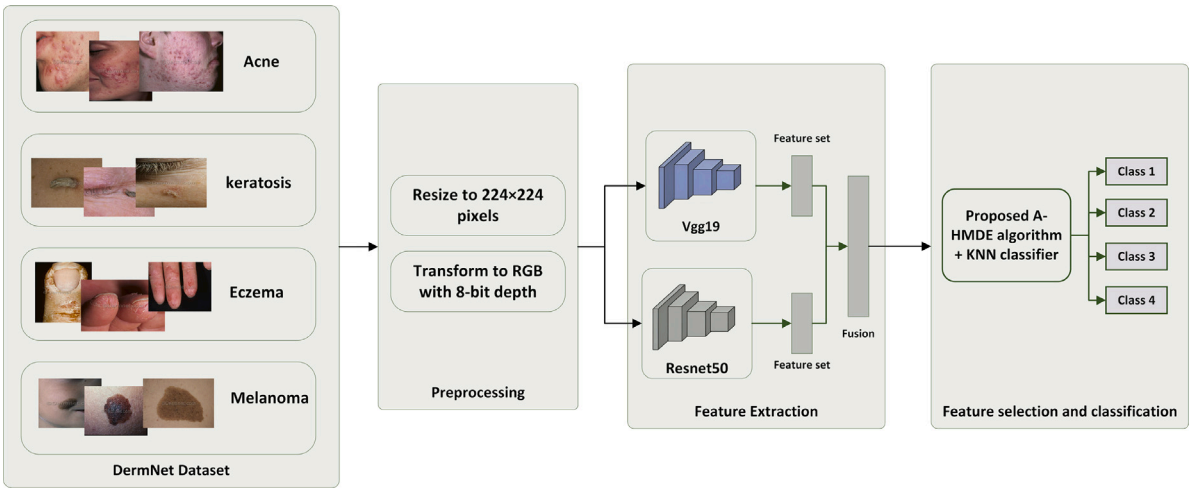
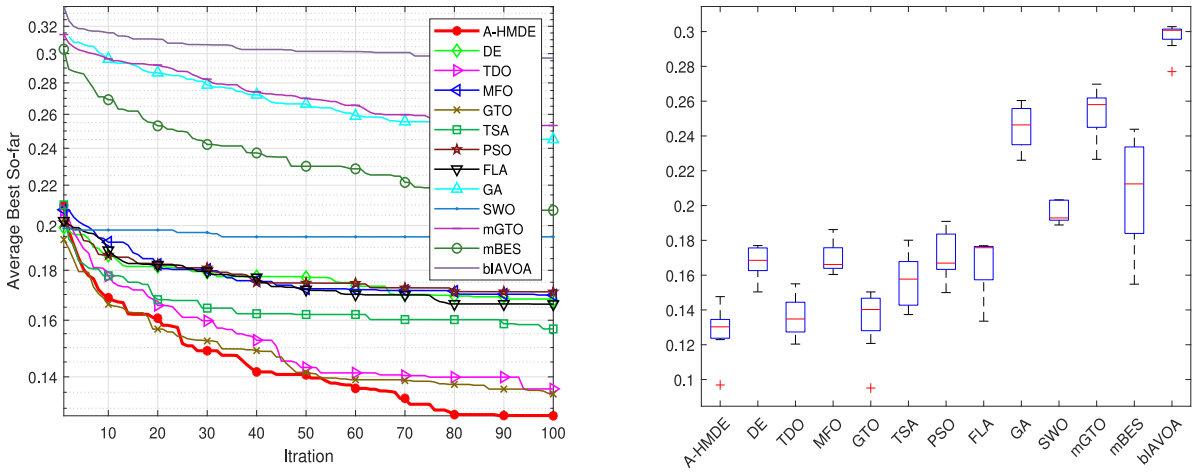


Fig. 11. Skin diseases prediction framework based on A-HMDE algorithm.



(a) Convergence curve (b) Boxplot

Fig. 12. Convergence curve for A-HMDE against other competitors — Skin diseases prediction.

Table 12
Skin diseases prediction results.

Algorithms	A-HMDE	DE	TDO	MFO	GTO	TSA	PSO	FLA	GA	SWO	mGTO	mBES	bIAVOA
Fitness values													
Mean	**0.127717**	0.16762	0.136066	0.169777	0.134453	0.156751	0.17107	0.166195	0.245149	0.19477	0.253249	0.207368	0.296911
STD	0.014662	0.009538	0.011594	0.008704	0.018509	0.015454	0.013883	0.015872	0.008608	**0.00598**	0.01391	0.031784	0.008727
Rank	1	6	3	7	2	4	8	5	11	9	12	10	13
Accuracy													
Mean	**0.87276**	0.8355	0.863518	0.833411	0.865315	0.84194	0.831601	0.835254	0.756559	0.80838	0.749142	0.792007	0.705666
STD	0.014516	0.009672	0.01136	**0.008804**	0.018387	0.015652	0.014045	0.015945	0.009082	0.00981	0.014083	0.030764	0.010008
Rank	1	5	3	7	2	4	8	6	12	9	11	10	13
Sensitivity													
Mean	**1**	0.99734	0.970238	**1**	0.982143	0.922619	**1**	0.994048	0.928571	0.99048	0.875	0.883929	0.839286
STD	**0**	0.007267	0.043625	**0**	0.03543	0.062022	**0**	0.016836	0.03818	0.02008	0.033065	0.065438	0.050508
Rank	1	2	6	1	5	8	1	3	7	4	10	9	11
Specificity													
Mean	**0.810096**	**0.810096**	0.771635	0.807692	0.78125	0.78125	0.793269	0.795673	0.743644	0.81923	0.754237	0.754237	0.724576
STD	**0.006799**	0.012324	0.029859	0.025179	0.027075	0.036973	0.019906	0.020397	0.010862	0.00993	0.027179	0.041517	0.029704
Rank	1	1	8	3	6	7	5	4	10	2	9	9	11
Feature size													
Mean	284.125	953	189.875	970.75	223	**54.25**	871	619.25	828.5	1012.2	979.875	291	1104
STD	27.84363	**18.15804**	181.0694	27.51493	132.6844	25.27986	23.26248	194.0668	86.01661	22.7928	23.81139	344.2731	515.0617
Rank	4	9	2	10	3	1	8	6	7	12	11	5	13

techniques in addressing the intricate task of skin disease prediction. The method's consistent capacity to achieve exceptional fitness values, coupled with sustained accuracy rates and precise FS, solidifies its status as a dependable and efficient strategy for confronting analogous challenges.

CRediT authorship contribution statement

Reham R. Mostafa: Writing – original draft, Visualization, Validation, Software, Methodology, Investigation, Formal analysis, Conceptualization. **Ahmed M. Khedr:** Writing – review & editing, Investigation, Validation, Supervision, Formal analysis, Conceptualization . **Zaher Al Aghbari:** Writing – review & editing, Investigation, Validation, Formal analysis, Supervision, Conceptualization. **Imad Afyouni:** Writing – review & editing, Visualization, Investigation, Data curation, Validation, Formal analysis. **Ibrahim Kamel:** Writing – review & editing, Validation, Formal analysis, Data curation. **Naveed Ahmed:** Writing – review & editing, Investigation, Validation, Formal analysis, Resources, Conceptualization, Supervision.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

All the data utilized in this paper are publicity available.

References

[1] Shaohua Wu, Yong Hu, Wei Wang, Xinyong Feng, Wanneng Shu, et al., Application of global optimization methods for feature selection and machine learning, Math. Probl. Eng. 2013 (2013).

[2] Fatemeh Behrad, Mohammad Saniee Abadeh, An overview of deep learning methods for multimodal medical data mining, Expert Syst. Appl. 200 (2022) 117006.

[3] Tansel Dokeroglu, Ayça Deniz, Hakan Ezgi Kiziloz, A comprehensive survey on recent metaheuristics for feature selection, Neurocomputing 494 (2022) 269–296.

[4] Erick Odhiambo Omuya, George Onyango Okeyo, Michael Waema Kimwele, Feature selection for classification using principal component analysis and information gain, Expert Syst. Appl. 174 (2021) 114765.

[5] Manoranjan Dash, Huan Liu, Feature selection for classification, Intell. Data Anal. 1 (1–4) (1997) 131–156.

[6] El-Sayed M. El-Kenawy, Abdelhameed Ibrahim, Seyedali Mirjalili, Marwa Metwally Eid, Sherif E. Hussein, Novel feature selection and voting classifier algorithms for COVID-19 classification in CT images, IEEE Access 8 (2020) 179317–179335.

[7] Manik Sharma, Prableen Kaur, A comprehensive analysis of nature-inspired meta-heuristic techniques for feature selection problem, Arch. Comput. Methods Eng. 28 (2021) 1103–1127.

[8] Girish Chandrashekar, Ferat Sahin, A survey on feature selection methods, Comput. Electr. Eng. 40 (1) (2014) 16–28.

[9] Razieh Sheikhpour, Mehdi Agha Sarram, Sajjad Gharaghani, Mohammad Ali Zare Chahooki, A survey on semi-supervised feature selection methods, Pattern Recognit. 64 (2017) 141–158.

[10] Maha Nssibi, Ghaith Manita, Ouajdi Korbaa, Advances in nature-inspired meta-heuristic optimization for feature selection problem: A comprehensive survey, Comp. Sci. Rev. 49 (2023) 100559.

[11] Sanat Jain, Ashish Jain, Mahesh Jangid, Review of metaheuristic techniques for feature selection, in: Soft Computing: Theories and Applications: Proceedings of SoCTA 2022, Springer, 2023, pp. 397–410.

[12] Reham R. Mostafa, Noha E. El-Attar, Sahar F. Sabbeh, Ankit Vidyarthi, Fatma A. Hashim, ST-AL: A hybridized search based metaheuristic computational algorithm towards optimization of high dimensional industrial datasets, Soft Comput. (2022) 1–29.

[13] Bing Xue, Mengjie Zhang, Will N. Browne, Xin Yao, A survey on evolutionary computation approaches to feature selection, IEEE Trans. Evol. Comput. 20 (4) (2015) 606–626.

[14] Jingwei Too, Seyedali Mirjalili, General learning equilibrium optimizer: A new feature selection method for biological data classification, Appl. Artif. Intell. 35 (3) (2021) 247–263.

[15] Gaurav Dhiman, Diego Oliva, Amandeep Kaur, Krishna Kant Singh, S. Vimal, Ashutosh Sharma, Korhan Cengiz, BEPO: A novel binary emperor penguin optimizer for automatic feature selection, Knowl.-Based Syst. 211 (2021) 106560.

[16] Mohammad Tubishat, Mohammed Alswaitti, Seyedali Mirjalili, Mohammed Ali Al-Garadi, Toqir A. Rana, et al., Dynamic butterfly optimization algorithm for feature selection, IEEE Access 8 (2020) 194303–194314.

[17] Farhad Soleimanian Gharehchopogh, Isa Maleki, Zahra Asheghi Dizaji, Chaotic vortex search algorithm: Metaheuristic algorithm for feature selection, Evol. Intell. 15 (3) (2022) 1777–1808.

[18] Mohammad H. Nadimi-Shahraki, Hoda Zamani, DMDE: Diversity-maintained multi-trial vector differential evolution algorithm for non-decomposition large-scale global optimization, Expert Syst. Appl. 198 (2022) 116895.

[19] Reham R. Mostafa, Marwa A. Gaheen, Mohamed Abd ElAziz, Mohammed Azmi Al-Betar, Ahmed A. Ewees, An improved gorilla troops optimizer for global optimization problems and feature selection, Knowl.-Based Syst. 269 (2023) 110462.

[20] Prachi Agrawal, Talari Ganesh, Ali Wagdy Mohamed, Chaotic gaining sharing knowledge-based optimization algorithm: An improved metaheuristic algorithm for feature selection, Soft Comput. 25 (14) (2021) 9505–9528.

[21] Gehad Ismail Sayed, Alaa Tharwat, Aboul Ella Hassanien, Chaotic dragonfly algorithm: An improved metaheuristic algorithm for feature selection, Appl. Intell. 49 (2019) 188–205.

[22] Zenab Mohamed Elgamal, Norizan Binti Mohd Yasin, Mohammad Tubishat, Mohammed Alswaitti, Seyedali Mirjalili, An improved harris hawks optimization algorithm with simulated annealing for feature selection in the medical field, IEEE Access 8 (2020) 186638–186652.

[23] Ahmed A. Ewees, Reham R. Mostafa, Rania M. Ghoniem, Marwa A. Gaheen, Improved seagull optimization algorithm using Lévy flight and mutation operator for feature selection, Neural Comput. Appl. 34 (10) (2022) 7437–7472.

[24] Rainer Storn, Kenneth Price, Differential evolution-a simple and efficient heuristic for global optimization over continuous spaces, J. Global Optim. 11 (4) (1997) 341.

[25] John H. Holland, Genetic algorithms, Sci. Am. 267 (1) (1992) 66–73.

[26] Rawaa Dawoud Al-Dabbagh, Azeddien Kinsheel, Saad Mekhilef, Mohd Sapiyan Baba, Shahaboddin Shamshirband, System identification and control of robot manipulator based on fuzzy adaptive differential evolution algorithm, Adv. Eng. Softw. 78 (2014) 60–66.

[27] Guoyue Luo, Lai Zou, Ziling Wang, Chong Lv, Jing Ou, Yun Huang, A novel kinematic parameters calibration method for industrial robot based on Levenberg-Marquardt and differential evolution hybrid algorithm, Robot. Comput.-Integr. Manuf. 71 (2021) 102165.

[28] Miao Qin, Rongbo Zhu, A Monte Carlo localization method based on differential evolution optimization applied into economic forecasting in mobile wireless sensor networks, EURASIP J. Wireless Commun. Networking 2018 (1) (2018) 1–9.

[29] Pratyay Kuila, Prasanta K. Jana, A novel differential evolution based clustering algorithm for wireless sensor networks, Appl. Soft Comput. 25 (2014) 414–425.

[30] Dezhi Han, Yunping Yu, Kuan-Ching Li, Rodrigo Fernandes de Mello, Enhancing the sensor node localization algorithm based on improved DV-HoP and DE algorithms in wireless sensor networks, Sensors 20 (2) (2020) 343.

[31] B.V. Babu, Ashish M. Gujarathi, Multi-Objective Differential Evolution (MODE) for optimization of supply chain planning and management, in: 2007 IEEE Congress on Evolutionary Computation, IEEE, 2007, pp. 2732–2739.

[32] Hao Guo, Gang Liu, Ying Zhang, Chunnan Zhang, Chuanhui Xiong, Wenli Li, A hybrid differential evolution algorithm for a location-inventory problem in a closed-loop supply chain with product recovery, Complex Intell. Syst. (2022) 1–23.

[33] Su Nguyen, Dhananjay Thiruvady, Andreas T. Ernst, Damminda Alahakoon, A hybrid differential evolution algorithm with column generation for resource constrained job scheduling, Comput. Oper. Res. 109 (2019) 273–287.

[34] Ying Hou, Yilin Wu, Honggui Han, Multi-objective differential evolution algorithm balancing multiple stakeholders for low-carbon order scheduling in E-waste recycling, IEEE Trans. Evol. Comput. (2023).

[35] Hamed Naseri, Mehrdad Ehsani, Amir Golroo, Fereidoon Moghadas Nejad, Sustainable pavement maintenance and rehabilitation planning using differential evolutionary programming and coyote optimisation algorithm, Int. J. Pav. Eng. 23 (8) (2022) 2870–2887.

[36] Zhenqian Xue, Shuo Yao, Haoming Ma, Chi Zhang, Kai Zhang, Zhangxin Chen, Thermo-economic optimization of an Enhanced Geothermal System (EGS) based on machine learning and differential evolution algorithms, Fuel 340 (2023) 127569.

[37] Akinola Ikudayisi, Josiah Adeyemo, John Odiyo, Abimbola Enitan, Optimum irrigation water allocation and crop distribution using combined Pareto multi-objective differential evolution, Cogent Eng. 5 (1) (2018) 1535749.

[38] Sotirios K. Goudos, Katherine Siakavara, Theodoros Samaras, Elias E. Vafiadis, John N. Sahalos, Self-adaptive differential evolution applied to real-valued antenna and microwave design problems, IEEE Trans. Antennas Propag. 59 (4) (2011) 1286–1298.

[39] Shi-Zheng Zhao, Ponnuthurai Nagaratnam Suganthan, Swagatam Das, Self-adaptive differential evolution with multi-trajectory search for large-scale optimization, Soft Comput. 15 (2011) 2175–2185.

[40] Yinzhi Zhou, Xinyu Li, Liang Gao, A differential evolution algorithm with intersect mutation operator, Appl. Soft Comput. 13 (1) (2013) 390–401.
[41] Wu Deng, Shifan Shang, Xing Cai, Huimin Zhao, Yingjie Song, Junjie Xu, An improved differential evolution algorithm and its application in optimization problem, Soft Comput. 25 (2021) 5277–5298.
[42] Mohammed Alswaitti, Mohanad Albughdadi, Nor Ashidi Mat Isa, Variance-based differential evolution algorithm with an optional crossover for data clustering, Appl. Soft Comput. 80 (2019) 1–17.
[43] Jirong Gu, Guojun Gu, Differential evolution with a local search operator, in: 2010 2nd International Asia Conference on Informatics in Control, Automation and Robotics, Vol. 2, CAR 2010, IEEE, 2010, pp. 480–483.
[44] Xueqi Wu, Ada Che, A memetic differential evolution algorithm for energy-efficient parallel machine scheduling, Omega 82 (2019) 155–165.
[45] Ashish Ghosh, Aloke Datta, Susmita Ghosh, Self-adaptive differential evolution for feature selection in hyperspectral image data, Appl. Soft Comput. 13 (4) (2013) 1969–1977.
[46] Uroš Mlakar, Iztok Fister, Janez Brest, Božidar Potočnik, Multi-objective differential evolution for feature selection in facial expression recognition systems, Expert Syst. Appl. 89 (2017) 129–137.
[47] T. Vivekanandan, N. Ch Sriman Narayana Iyengar, Optimal feature selection using a modified differential evolution algorithm and its effectiveness for prediction of heart disease, Comput. Biol. Med. 90 (2017) 125–136.
[48] Subrat Kumar Nayak, Pravat Kumar Rout, Alok Kumar Jagadev, Tripti Swarnkar, Elitism-based multi-objective differential evolution with extreme learning machine for feature selection: A novel searching technique, Connect. Sci. 30 (4) (2018) 362–387.
[49] Le Yao, Zhiqiang Ge, Variable selection for nonlinear soft sensor development with enhanced binary differential evolution algorithm, Control Eng. Pract. 72 (2018) 68–82.
[50] Emrah Hancer, Fuzzy kernel feature selection with multi-objective differential evolution algorithm, Connect. Sci. 31 (4) (2019) 323–341.
[51] Yong Zhang, Dun-wei Gong, Xiao-zhi Gao, Tian Tian, Xiao-yan Sun, Binary differential evolution with self-learning for multi-objective feature selection, Inform. Sci. 507 (2020) 67–85.
[52] Rafael Rivera-López, Efrén Mezura-Montes, Juana Canul-Reich, Marco Antonio Cruz-Chávez, A permutational-based differential evolution algorithm for feature subset selection, Pattern Recognit. Lett. 133 (2020) 86–93.
[53] Jeng-Shyang Pan, Nengxian Liu, Shu-Chuan Chu, A competitive mechanism based multi-objective differential evolution algorithm and its application in feature selection, Knowl.-Based Syst. 245 (2022) 108582.
[54] Emrah Hancer, Bing Xue, Mengjie Zhang, Fuzzy filter cost-sensitive feature selection with differential evolution, Knowl.-Based Syst. 241 (2022) 108259.
[55] Emrah Hancer, Bing Xue, Mengjie Zhang, An evolutionary filter approach to feature selection in classification for both single-and multi-objective scenarios, Knowl.-Based Syst. (2023) 111008.
[56] Suchitra Agrawal, Aruna Tiwari, Bhaskar Yaduvanshi, Prashant Rajak, Feature subset selection using multimodal multiobjective differential evolution, Knowl.-Based Syst. 265 (2023) 110361.
[57] Yanmei Hu, Min Lu, Xiangtao Li, Biao Cai, Differential evolution based on network structure for feature selection, Inform. Sci. 635 (2023) 279–297.
[58] Shichao Zhang, Xuelong Li, Ming Zong, Xiaofeng Zhu, Ruili Wang, Efficient KNN classification with different numbers of nearest neighbors, IEEE Trans. Neural Netw. Learn. Syst. 29 (5) (2017) 1774–1785.
[59] Mohamed Abdel-Basset, Reda Mohamed, Mohammed Jameel, Mohamed Abouhawwash, Spider wasp optimizer: A novel meta-heuristic optimization algorithm, Artif. Intell. Rev. (2023) 1–64.
[60] Iman Ahmadianfar, Ali Asghar Heidari, Amir H. Gandomi, Xuefeng Chu, Huiling Chen, RUN beyond the metaphor: An efficient optimization algorithm based on runge kutta method, Expert Syst. Appl. 181 (2021) 115079.
[61] Mohammad Dehghani, Štěpán Hubálovskỳ, Pavel Trojovskỳ, Tasmanian devil optimization: A new bio-inspired optimization algorithm for solving optimization algorithm, IEEE Access 10 (2022) 19599–19620.
[62] Seyedali Mirjalili, Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm, Knowl.-Based Syst. 89 (2015) 228–249.
[63] Benyamin Abdollahzadeh, Farhad Soleimanian Gharehchopogh, Seyedali Mirjalili, Artificial gorilla troops optimizer: A new nature-inspired metaheuristic algorithm for global optimization problems, Int. J. Intell. Syst. 36 (10) (2021) 5887–5958.
[64] Satnam Kaur, Lalit K. Awasthi, A.L. Sangal, Gaurav Dhiman, Tunicate swarm algorithm: A new bio-inspired based metaheuristic paradigm for global optimization, Eng. Appl. Artif. Intell. 90 (2020) 103541.
[65] Russell Eberhart, James Kennedy, Particle swarm optimization, in: Proceedings of the IEEE International Conference on Neural Networks, Vol. 4, Citeseer, 1995, pp. 1942–1948.
[66] Fatma A. Hashim, Reham R. Mostafa, Abdelazim G. Hussien, Seyedali Mirjalili, Karam M. Sallam, Fick's Law Algorithm: A physical law-based algorithm for numerical optimization, Knowl.-Based Syst. 260 (2023) 110146.
[67] Zakieh Sharifian, Behrang Barekatain, Alfonso Ariza Quintana, Zahra Beheshti, Faramarz Safi-Esfahani, Sin-Cos-bIAVOA: A new feature selection method based on improved african vulture optimization algorithm and a novel transfer function to DDoS attack detection, Expert Syst. Appl. 228 (2023) 120404.
[68] Amit Chhabra, Abdelazim G. Hussien, Fatma A. Hashim, Improved bald eagle search algorithm for global optimization and feature selection, Alex. Eng. J. 68 (2023) 141–180.
[69] Arthur Asuncion, David Newman, UCI Machine Learning Repository, Irvine, CA, USA, 2007.
[70] Joaquín Derrac, Salvador García, Daniel Molina, Francisco Herrera, A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms, Swarm Evol. Comput. 1 (1) (2011) 3–18.
[71] Karen Simonyan, Andrew Zisserman, Very deep convolutional networks for large-scale image recognition, 2014, arXiv preprint arXiv:1409.1556.
[72] Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 770–778.