Advances in Industrial Engineering, December 2025, 59(2): 451-469
DOI: 10.22059/aie.2025.388689.1936

RESEARCH PAPER

Dual-Phase Local Search Embedded Multi-Verse Optimizer for Optimal Feature Subset Selection

Maryam Askari[1], Farid Khoshalhan[2 *] and Hodjat Hamidi[2]
[1]*Ph.D. Candidate, Department of Industrial Engineering, K. N. Toosi University of Technology, Tehran, Iran.*
[2]*Associate Professor, Department of Industrial Engineering, K.N. Toosi University of Technology, Tehran, Iran.*

Received: 16 January 2025, Revised: 26 August 2025, Accepted: 28 August 2025

Abstract

Feature selection plays a pivotal role in enhancing the performance of machine learning models by reducing dimensionality, improving interpretability, and minimizing computational overhead. This study presents the Improved Multi-Verse Optimizer (IMVO), a new feature selection algorithm that merges the global exploration ability of the standard Multi-Verse Optimizer (MVO) with a dual-phase mutation-based local search mechanism. Unlike previous MVO-based or hybrid metaheuristic approaches, IMVO simultaneously strengthens exploitation through targeted refinement of the best solutions and preserves population diversity via periodic random-solution mutations. This strategic combination mitigates premature convergence, accelerates convergence speed, and improves robustness across diverse high-dimensional datasets. Comprehensive experiments on 14 widely used datasets obtained from the UCI repository show that IMVO consistently achieves higher classification accuracy, fewer selected features, and lower fitness values than five state-of-the-art algorithms (MVO, GA, PSO, SSA, HHO). Quantitative analysis using the Wilcoxon signed-rank test certifies the significance of these enhancements, underscoring the algorithm's reliability. While the inclusion of local search increases computational cost, the demonstrated gains in accuracy, stability, and feature reduction affirm this cost-benefit relationship, positioning IMVO as a competitive and versatile tool for feature selection and related optimization problems.

Keywords:
Feature Selection, Multi-Verse Optimizer, Optimization Algorithm, Local Search Enhancement.

Introduction

Feature or variable selection is an essential step in data preprocessing, especially when handling high-dimensional datasets. The goal of feature selection is to reduce the data's dimensionality, improving the performance of machine learning models. By enhancing the accuracy of predictive models, mitigating overfitting, accelerating training, and simplifying models for better interpretability, feature selection plays a crucial role. Its significance is particularly evident in scenarios where training data is limited and irrelevant or redundant features can degrade model performance [1]. Moreover, in domains like bioinformatics, document analysis, intrusion detection, and financial analytics where obtaining features can be resource-intensive identifying a minimal but highly informative subset of features is imperative. Additionally, feature selection aids in better data comprehension by identifying key variables, thus supporting informed decision-making.

To achieve high classification accuracy while reducing dimensionality, the feature selection

*Corresponding author: (Farid Khoshalhan)
Email: khoshalhan@kntu.ac.ir

process typically comprises assessing subsets of features, exploiting search techniques to locate optimal subsets, assessing the selected features, defining stopping criteria, and validating the results [2]. Feature selection methods are generally categorized into three approaches: filter, wrapper, and embedding methods. Filter methods, which are independent of specific learning algorithms, assess features based on intrinsic properties and are efficient for large datasets due to their scalability. Common examples incorporate Fisher Score, Correlation Coefficient, Information Gain, Variance Threshold, and Chi-Square. Wrapper methods, on the other hand, use learning algorithms to evaluate subsets' performance within a predictive model's context. While these methods often achieve higher accuracy by considering feature interactions with the model, they can be computationally intensive and prone to overfitting. Popular wrapper techniques include decision trees, support vector machines, and nearest neighbors [3]. Embedded methods merge feature selection directly into the model training process, balancing accuracy and computational efficiency. Notable examples comprise Lasso regression and tree-based methods like Gradient Boosting Machines and Random Forests. The choice of an evaluation method depends on the specific requirements and constraints of the problem, as each has distinct advantages and limitations [4].

Embedded methods like L1-regularized models and tree-based feature importance perform well in certain settings but are classifier-specific, potentially biased, and less suitable for heterogeneous datasets. Filter methods such as Mutual Information and Relief-F are efficient and model-agnostic but often miss higher-order feature interactions, leading to redundancy in complex, high-dimensional. Deep feature selection with neural networks or autoencoders can be effective for large-scale data but requires extensive tuning, large sample sizes, and high computational resources, while offering limited interpretability making them less practical for smaller datasets.

Given a dataset with n features, there are 2n possible feature subsets [5]. Exhaustively evaluating all subsets is computationally infeasible, prompting the use of heuristic-driven random searches to find satisfactory, albeit suboptimal, solutions. Recent years have seen a growing interest in advanced heuristic and metaheuristic methods for feature selection across various domains [6], including clinical diagnosis, engineering, text mining, intrusion detection, and security applications [7]. These approaches effectively navigate complex solution spaces, identifying practical feature subsets within feasible computational restrictions.

The "No Free Lunch" (NFL) theorem states that no single optimization algorithm universally outperforms others for all problems [8]. Consequently, researchers are motivated to develop new algorithms or enhance existing ones. Among various metaheuristic algorithms, the Multi-Verse Optimizer (MVO) has proven effective. Inspired by physics concepts like wormholes, white holes, and black holes, MVO explores the solution space by simulating the expansion of multiple universes. Its exploration capabilities make it well-suited for complex problems like feature selection. However, like other evolutionary algorithms, MVO can endure from slow convergence and local optima. Local search algorithms (LSAs), by contrast, refine solutions by intensifying searches around promising regions. When combined with MVO, LSAs complement its global search capability by fine-tuning promising solutions, improving overall optimization performance through a balance of exploration and exploitation.

Despite the promising exploration capability of MVO, prior studies [17] highlight several persistent limitations:

1. Premature convergence in the later search phases due to loss of population diversity.
2. Slow convergence speed in refining promising solutions.
3. Sensitivity to initial population distribution, which can hinder robust performance on high-dimensional datasets.

Hybrid approaches, such as MVO combined with Simulated Annealing or Opposition-Based Learning, have alleviated some of these issues, but often at the cost of either increased

computational load without proportional accuracy gains or inadequate balance between exploration and exploitation. In particular, most existing hybrids do not provide a systematic mechanism to enhance local search precision while simultaneously preserving diversity.

To address these shortcomings, this study integrates a novel, mutation-based Local Search Algorithm (LSA) directly within the iterative process of MVO. The integration is specifically designed to (i) fine-tune the best solution through targeted mutations, improving exploitation efficiency, and (ii) periodically mutate randomly selected solutions, injecting diversity into the population to avoid stagnation. These two stages operate in a complementary fashion, ensuring that promising regions of the search space are explored more thoroughly without sacrificing global coverage.

Our proposed method innovatively integrates LSAs with MVO for feature selection. While MVO excels in global exploration and identifying promising feature subsets, LSAs refine these subsets to achieve optimal solutions. This two-phase approach enhances feature selection effectiveness and reduces computational costs by concentrating efforts on high-potential regions. The synergy between MVO's global search and LSAs' local refinement addresses their individual limitations, making the hybrid approach particularly advantageous for high-dimensional datasets with vast, complex search spaces. This study makes the following key contributions:

1. Development of the Improved Multi-Verse Optimizer (IMVO) algorithm, an enhanced MVO variant for wrapper-mode feature selection.
2. Overview of a novel LSA mechanism to boost population diversity and exploitation in the IMVO algorithm through random selection and mutation.
3. Dual functionality of the LSA mechanism: enhancing solution diversity and improving the best solution using mutation, thus mitigating local optima issues.
4. Comparative evaluation of IMVO against five popular optimization methods through extensive tests on 14 UCI datasets with varying dimensionalities.

The structure of the paper is as follows: Section 2 reviews related work and previous studies. Section 3 provides a detailed explanation of the MVO algorithm, including its mathematical foundation. Section 4 introduces our proposed hybrid algorithm, highlighting its distinctive features and implementation. Section 5 presents and analyzes the experimental results, emphasizing the method's performance on various datasets. Finally, Section 6 provides a summary of the conclusions and highlights potential directions for future research.

Literature Works

Feature selection is considered one of the key stages in machine learning and knowledge extraction, as eliminating irrelevant features and selecting an optimal subset can not only improve the accuracy of classification models but also reduce computational complexity and enhance the interpretability of results [9]. This becomes particularly important in high-dimensional datasets, where the presence of redundant features may lead to the curse of dimensionality and reduce model performance.

Given this challenge, recent studies have increasingly focused on developing metaheuristic algorithms for feature selection, as these algorithms are capable of exploring large and complex search spaces and achieving high-quality solutions [9]. In general, metaheuristic algorithms in this domain are divided into two main categories: population-based and path-based algorithms. While path -based approaches, such as Tabu Search and Simulated Annealing (SA), iteratively improve a single solution and are often prone to getting trapped in local optima, population-based algorithms, such as the Genetic Algorithm (GA), Whale Optimization Algorithm (WOA), and Cuckoo Search (CS), leverage multiple search agents, thus providing greater diversity and a stronger ability to perform global exploration [9].

In recent years, there has been a growing interest in designing hybrid algorithms that combine the global search capabilities of one metaheuristic with the local exploitation strength of another algorithm, thereby optimizing both exploration and exploitation simultaneously. For instance, integrating Bee Colony optimization with Gradient Boosted Decision Trees has significantly improved classification accuracy and reduced data dimensionality in medical datasets [10]. Similarly, the Chaotic Harris Hawks Optimization (CHHO) algorithm, which incorporates chaotic maps to enhance population diversity and uses Simulated Annealing (SA) during the exploitation phase, has demonstrated superior performance in both feature selection and classification accuracy compared to HHO and other metaheuristics [11].

In line with improving the balance between local and global search, the Improved Salp Swarm Algorithm (ISSA) integrates a mutation-based Local Search Algorithm (LSA) and Opposition-Based Learning (OBL) to simultaneously enhance population diversity and effective exploitation. Experimental results on UCI datasets show that ISSA excels its baseline algorithms in terms of feature reduction, classification accuracy, and fitness values [12]. Similarly, the Henry Gas Solubility Optimization (HGSO) algorithm has proven effective in dimensionality reduction and improving classification accuracy, especially for high-dimensional datasets [13].

Meanwhile, some studies have focused on improving binary versions of classical algorithms. For example, the Binary Grey Wolf Optimizer (BGWO), by modifying the parameters of the standard GWO model, has shown significant improvements in convergence speed and solution quality for feature selection problems [14]. Likewise, the Multi-Population Particle Swarm Optimization (MPPSO) algorithm, which combines random initialization with feature ranking based on ReliefF, achieves higher classification accuracy on high-dimensional datasets compared to conventional methods [15]. Moreover, for Intrusion Detection Systems (IDS), a novel pigeon inspired optimization based method has been proposed. By employing an innovative binarization strategy, this approach outperforms existing techniques in terms of True Positive Rate (TPR), False Positive Rate (FPR), and F1-score [16].

Despite significant advancements in feature selection algorithms, the No Free Lunch (NFL) theorem [8] states that no algorithm performs best across all problems. This limitation has prompted researchers to increasingly focus on hybrid algorithms in recent years. These approaches aim to combine the strengths of global exploration with precise local exploitation, thereby mitigating the weaknesses of individual methods.

For example, in job-shop scheduling problems, combining the Whale Optimization Algorithm (WOA) with local search has produced remarkable results. This hybrid not only improves solution quality but also guides the search toward promising regions, thereby avoiding premature convergence to local optima [17]. Continuing along this path, the Dynamic Butterfly Optimization Algorithm (DBOA) integrates a mutation-based local search algorithm, significantly enhancing population diversity and accelerating convergence. Experimental results on UCI datasets show that DBOA outperforms its standard counterpart in both feature selection and dimensionality reduction [18].

Similarly, in energy demand forecasting, integrating the Ant Colony Optimization (ACO) algorithm with Iterated Local Search (ILS) has demonstrated that combining global exploration with targeted local refinement can significantly improve prediction accuracy [19]. In the field of biomedical data, the Binary Coral Reef Optimization with Simulated Annealing (BCROSAT) algorithm has notably enhanced diagnostic accuracy and stability compared to traditional approaches [20].

Finally, combining Cuckoo Search with Hill Climbing (CSAHC) represents another step forward. By establishing an optimal balance between global exploration and local exploitation, this hybrid achieves better performance than the standard CS and improves efficiency in solving complex optimization problems [21].

A review of these studies highlights that hybrid metaheuristic-based approaches are highly effective in tackling the challenges of high-dimensional feature selection problems. However, most existing hybrid algorithms still face several critical challenges: Premature convergence during the early stages of the search, Gradual loss of population diversity over iterations, Insufficient exploitation of promising regions within the search space.

Inspired by these limitations, this study introduces a novel algorithm called the Improved Multi-Verse Optimizer (IMVO). By combining the global exploration ability of the Multi-Verse Optimizer (MVO) [22] with a mutation-based Local Search Algorithm (LSA) [23], IMVO adopts a two-phase approach to achieve an optimal balance between exploration and exploitation. In this framework, the first phase leverages MVO to identify promising regions in the search space, while the second phase applies LSA to refine these regions more effectively. Furthermore, incorporating random mutations during different stages of the search helps maintain population diversity and prevents premature convergence.

Research Approach

MVO Inspiration

The Big Bang theory and the Multiverse theory are foundational concepts in physics that inspired the development of the Multi-Verse Optimizer (MVO) algorithm [22]. According to the Big Bang theory, the universe as we know it originated from an immense explosion, marking the beginning of time and space. Prior to this event, there was no presence. On the other hand, the multiverse theory suggests the presence of multiple universes, each arising from its own version of the Big Bang. These universes may differ significantly, possessing unique sets of physical laws, and can interact or collide with one another.

Three core concepts from the multiverse hypothesis white holes, black holes, and wormholes played a pivotal role in shaping the MVO algorithm:

- **White Holes:** Hypothetical counterparts to black holes, white holes are theorized to eject matter and energy rather than absorb it. While they have not been observed in our universe, some interpretations suggest they could represent the Big Bang itself. In the cyclic multiverse model, white holes are thought to form at points of interaction between parallel universes.
- **Black Holes:** Well-documented in our universe, these objects possess immense gravitational forces, capable of pulling in everything around them, including light.
- **Wormholes:** Imagined as tunnels through space-time, wormholes are theoretical constructs that could connect distant points within a universe or even link separate universes, enabling instantaneous travel over vast distances.

The MVO algorithm is inspired by cosmological concepts, modeling the interaction of multiple universes through mechanisms analogous to wormholes, black holes, and white holes. By leveraging these ideas, the algorithm models the process both intuitively and mathematically, aiming to achieve a stable and optimized state. This conceptual framework is explored in greater detail in subsequent sections.

MVO Algorithm

The MVO algorithm [22] categorizes its search process into two fundamental phases: exploration and exploitation. While the concept of wormholes underpins the exploitation phase, exploration is guided by the principles of white and black holes. In the MVO model, each solution is treated as a universe, and each variable within a solution represents an object within that universe. Solutions are assigned inflation rates, which correspond to the fitness function values. To align with the terminology of multiverse theory, the term "iteration" is replaced by "time."

The interaction among universes during the optimization process follows specific rules: higher inflation rates lead to an increased likelihood of white holes, while black hole probabilities decrease. Despite inflation rates, objects in all universes can randomly move through wormholes toward the best universe. Universes with higher inflation rates are more likely to emit objects via white holes, while those with lower inflation rates are more likely to receive objects via black holes.

Objects move between universes through tunnels that connect white holes and black holes. When such a tunnel is established, a white hole is identified in the universe with a higher inflation rate, while a black hole is located in the universe with a lower inflation rate. This process facilitates the transfer of objects from the white holes of the source universe to the black holes of the destination universe, enabling inter-universe exchange.

Universes with higher inflation rates tend to contain more white holes, whereas lower inflation rates are associated with black holes. This mechanism helps increase the overall inflation rate across all universes, enhancing the likelihood of object transfer from fast-inflating universes.

The movement of objects through these tunnels is modeled mathematically using a roulette wheel selection method. In this approach, universes are ranked based on their inflation rates, and one universe is selected as the white hole in each cycle. A matrix U is used to represent this process, where n represents the number of universes and d represents the parameters.

$$U = \begin{bmatrix} x_1^1 & x_1^2 & \cdots & x_1^d \\ x_2^1 & x_2^2 & \cdots & x_2^d \\ \vdots & \vdots & \vdots & \vdots \\ x_n^1 & x_n^2 & \cdots & x_n^d \end{bmatrix}$$

In Equation (1), Ui represents the i-th universe, NI(Ui) denotes its normalized inflation rate, x_i^j corresponds to the j-th parameter of the i-th universe, and r1 is a randomly generated value within the range [0,1]. The roulette wheel selection method is employed to determine the parameters x_k^j for the chosen universe.

This method ensures that universes with lower inflation rates, which are more likely to facilitate object transfers through tunnels, are given priority. For maximization problems, adjustments such as replacing −NI with NI are implemented. These mechanisms enhance exploration within the search domain by promoting object transfers and sudden shifts between universes.

$$x_i^j = \begin{cases} x_k^j & \text{r1} < \text{NI}(U_i) \\ x_i^j & \text{r1} \geq \text{NI}(U_i) \end{cases} \quad (1)$$

This section corresponds to parts (A and B) of the flowchart illustrated in Fig. 1. The roulette wheel approach governs the transfer of objects between universes through tunnels connecting white and black holes. This strategy is designed to preserve diversity while also enabling the potential for increased inflation rates. To promote broader exploration, each universe is equipped with its own wormholes, which facilitate the random transportation of objects across space. Unlike tunnels, which are influenced by inflation rates, wormholes introduce random alterations to objects within universes. These wormholes establish consistent links between a given universe and the best-performing universe identified so far, enabling local refinements and potentially enhancing inflation rates. The specifics of this mechanism are outlined in Equation (2).

$$x_i^j = \begin{cases} x_j + \text{TDR} \times ((\text{ub}_j - \text{lb}_j) \times \text{r4} + \text{lb}_j) \\ x_j - \text{TDR} \times ((\text{ub}_j - \text{lb}_j) \times \text{r4} + \text{lb}_j) \\ x_i^j \end{cases} \quad (2)$$

As illustrated in part (B) of Fig. 1, Xj represents the j-th parameter of the best-performing universe, while TDR and WEP are coefficients. The bounds of the j-th variable are denoted as lbj (lower bound) and ubj (upper bound). The i-th universe's j-th parameter is denoted by x_i^j, with r2, r3, and r4 being random values between 0 and 1.

This approach facilitates precise adjustments within universes by leveraging wormholes to exchange objects and strategically improve inflation rates. From the flowchart and mathematical expressions, two main coefficients emerge: the wormhole existence probability (WEP) and the traveling distance rate (TDR). WEP defines the probability of wormholes forming in universes and increases linearly throughout iterations, enhancing exploitation as optimization proceeds. TDR, on the other hand, determines the range of distances objects can travel through wormholes to reach the best-known universe. Unlike WEP, TDR rises over iterations to support more precise local exploration around the optimal universe. The adaptive formulas for these coefficients are provided in Equations (3) and (4), where their values range between 0.2 (minimum) and 1 (maximum). Here, l represents the current iteration, and L is the total number of iterations.

The parameter p, set to 6 in this study, regulates the precision of exploitation, with larger values allowing for more accurate local searches earlier in the process. The study suggests using adaptive values for WEP and TDR, even though constant values can also be employed.

$$WEP = \min + l \times \left(\frac{\max - \min}{L} \right) \quad (3)$$

$$TDR = 1 - \frac{l^{\frac{1}{p}}}{L^{\frac{1}{p}}} \quad (4)$$

The MVO algorithm initializes by generating random universes and employs white and black holes to transfer objects from universes with higher inflation rates to those with lower rates at the end of each iteration. Additionally, random wormhole teleportations to the best-performing universe occur concurrently. This iterative process continues until a specified stopping criterion, such as the maximum number of iterations, is reached.

The computational complexity of the algorithm depends on factors such as the number of universes, the sorting of these universes, the roulette wheel selection method, and the total iterations.

Universes are sorted using the Quicksort algorithm, which has a worst-case time complexity of O(n2) and a best-case complexity of O (n logn). The roulette wheel selection mechanism exhibits a complexity of either O(n) or O(logn). Consequently, the overall computational complexity is determined by Equations (5) and (6), where l represents the maximum number of iterations, d indicates the number of objects, and n denotes the total number of universes. The pseudocode of MVO algorithm is shown in Fig. 2.

$$O(MVO) = O(l(O(Quick\ \ sort) + n \times d \times (O(roulette\ \text{wheel})))) \quad (5)$$

$$O(MVO) = O(l(n^2 + n \times d \times \log n)) \quad (6)$$

The Refined Local Search Algorithm (LSA)

Figure 1 demonstrates how the exploitation capability of the MVO algorithm has been enhanced in this study by incorporating a refined LSA [23]. The LSA operates in two main ways: it either improves the current best solution or optimizes a randomly selected solution. If the randomly chosen solution is enhanced, LSA updates it accordingly and checks whether it surpasses the current best solution.

As shown in Fig. 3 the streamlined LSA process can be outlined as follows:

1. The best solution at the end of each MVO cycle, XBest, is passed to LSA.
2. The algorithm executes for a fixed number of iterations (Nit) to search for a superior solution.
3. During each iteration:

- LSA mutates XBest to produce a new candidate solution, Pnew.
- If the fitness value of Pnew (denoted as Nfit) exceeds that of XBest (denoted as Bfit), XBest is updated to Pnew.
- If Pnew does not improve XBest, LSA randomly selects another solution, Sselected, from the population.
- Sselected is mutated by LSA to generate a new solution, Pnew .
- If Nfit is greater than the fitness value of Sselected (denoted as Pfit), Sselected is exchanged with Pnew.
- If Sselected is enhanced, LSA compares Nfit to Bfit. If Bfit is superior, XBest is updated accordingly.

By following this procedure, LSA enhances MVO's ability to find better solutions while maintaining population diversity through random selection. The pseudocode of IMVO algorithm is shown in Fig. 4.

Experimental Result and Discussion

The datasets and parameter settings used in this study are described in this section. It also gives a thorough analysis of the results of the experiment and summarizes the results.

Datasets

Table 1 provides a summary of the datasets utilized in this study. To assess the effectiveness of the proposed IMVO algorithm, 14 benchmark datasets from the UCI repository were used. These datasets vary significantly in size and subject matter, with features ranging from 13 to 451 and instances spanning between 101 and 20,000. The diversity in dataset range, topics, and dimensions enhances the reproducibility and generalizability of our research. This wide variety ensures that the results are not confined to specific cases but are applicable to a broad spectrum of scenarios.

Table 1. Summary of 14 Benchmark Datasets Utilized for Assessing the IMVO Algorithm

#	Datasets	Feature count	Instance count
1	Breast cancer	30	569
2	Musk 1	168	476
3	Wine	13	178
4	Kidney disease	24	400
5	Sonar	60	208
6	Zoo	16	101
7	Ionosphere	34	351
8	German credit data	20	1000
9	Mice protein representation	82	1080
10	Darwin	451	174
11	Letter recognition	16	20000
12	Tennis match	42	127
13	Credit approval	15	690
14	Handwritten digits	64	5620

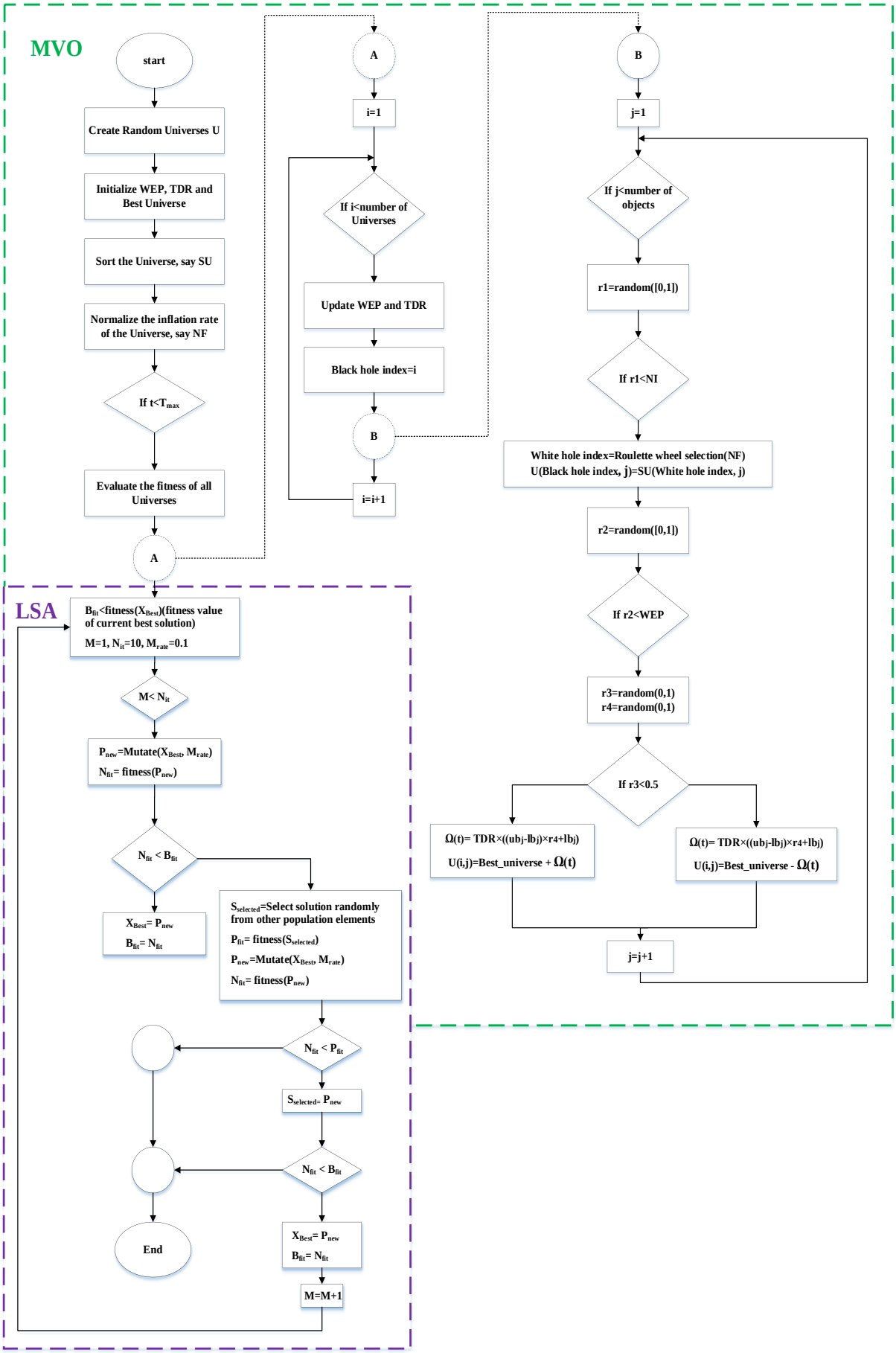


Figure 1. The Flowchart of the Hybrid IMVO Algorithm

```
Create random universes (U)
Initialize WEP, TDR, and Best_universe
SU = Sorted universes
NI = Normalize the inflation rate (fitnesses) of the universes
while the end criterion is not satisfied:
   Evaluate the fitness of all universes
   for each universe indexed by i:
      Update WEP and TDR
      Black_hole_index = i
      for each object indexed by j:
         r1 = random([0, 1])
         if r1 < NI(U[i]):
            White_hole_index = RouletteWheelSelection(-NI)
            U [Black_hole_index, j] = SU [White_hole_index, j]
         end if
         r2 = random([0, 1])
         if r2 < Wormhole_existance_probability:
            r3 = random([0, 1])
            r4 = random([0, 1])
            if r3 < 0.5:
               U[i, j] = Best_universe[j] + TDR * (ub[j] - lb[j]) * r4 + lb[j]
            else:
               U[i, j] = Best_universe[j] - TDR * (ub[j] - lb[j]) * r4 + lb[j]
            end if
         end if
      end for
   end for
end while
```

Figure 2. Pseudocode of MVO algorithm

```
B_fit = fitness(X_best)     # fitness value of current best solution
M = 1
N_it = 10
M_rate = 0.1
while M < N_it:
   P_new = Mutate (X_best, M_rate)
   N_fit = fitness(P_new)

   if N_fit < B_fit:
      X_best = P_new
      B_fit = N_fit
   else:
      S_selected = Select solution randomly from other population elements
      P_fit = fitness(S_selected)

      P_new = Mutate (S_selected, M_rate)
      N_fit = fitness(P_new)

      if N_fit < P_fit:
         S_selected = P_new

         if N_fit < B_fit:
            X_best = P_new
            B_fit = N_fit
```

```
        end if
      end if
   end if


   M = M + 1
end while
```

Figure 3. Pseudocode of LSA algorithm

```
t = 1          # Counter for number of iterations
Create random universes (U)
initialize WEP, TDR, Best_universe
SU = sorted_universes(U)
NI = normalize_inflation_rate(fitness) of the universes
t = t + 1
while t <= Tmax:
   Evaluate the fitness of all universes
   for each universe indexed by i:
      Update WEP and TDR
      Black_hole_index = i
      for each object indexed by j:
         r1 = random. ([0,1])
         if r1 < NI(U[i]):
            White_hole_index = roulette_wheel_selection(-NI)
            U[Black_hole_index,j] = SU[White_hole_index,j]
         end if
         r2 = random.([0,1])
         if r2 < Wormhole_existence_probability:
            r3 = random.([0,1])
            r4 = random.([0,1])
            if r3 < 0.5:
               U[i][j] = Best_universe[j] + TDR * (ub[j] - lb[j]) * r4 + lb[j]
            else:
               U[i][j] = Best_universe[j] - TDR * (ub[j] - lb[j]) * r4 + lb[j]
         end if
      end if
   end for
end for
Apply LSA
end while
return X_best
```

Figure 4. Pseudocode of IMVO algorithm

Defining Parameters

To benchmark the proposed IMVO algorithm, we compared it with five optimization algorithms: the original MVO, GA, SSA, HHO, and PSO. Consistent parameter settings were applied across all experiments to ensure fairness. The evaluation of optimal solutions was performed using the KNN method with k=5 and Euclidean distance. The value of k was determined through trial and error, with k=5 yielding the best results across all datasets. This approach was adapted from methodologies presented in [24], [25], [26]. We employed K-fold cross-validation, where datasets were divided into K segments based on the procedure described in [27]. For each fold, K−1 segments were used for training, while the remaining fold served as the testing set [28]. In this study, fivefold cross-validation was used, with one-fold allocated for

testing and four folds for training. The experiments utilized a population size of 10 and 100 iterations, aligning with the standard practices in feature selection research found in [10], [29], [30], [31], [32], [33]. Each algorithm was executed independently 30 times for every test. Table 3 provides the specific parameter values for each method, while Table 2 summarizes the overall parameter settings applied to all algorithms.

Table 2. Experimental Parameter Setup [34]

Parameters	Value
Algorithm's run number	30
Iteration number	100
Population	10
KFOLD	5
KNN	k = 5 and Euclidean distance

Table 3. Optimization Algorithm Parameter Setting

Algorithm	Parameter setting
PSO	Acceleration constants (C1 =2, C2 =2) [35], [36], [37], [38] Inertia Weight (W= 0.9)
MVO	-
SSA	-
HHO	-
GA	Mutation ratio = 0.1[36], [39], [40] Crossover ratio = 0.9
IMVO	Mutation rate = 0.1[34] Number of LSA iteration = 10

Findings and Interpretation

This section presents the evaluation results for each algorithm, based on 30 independent runs across 14 datasets. Various performance metrics are employed to compare the proposed IMVO algorithm with existing approaches. The results highlight the effectiveness and superior performance of IMVO, demonstrating its ability to achieve high-quality outcomes.

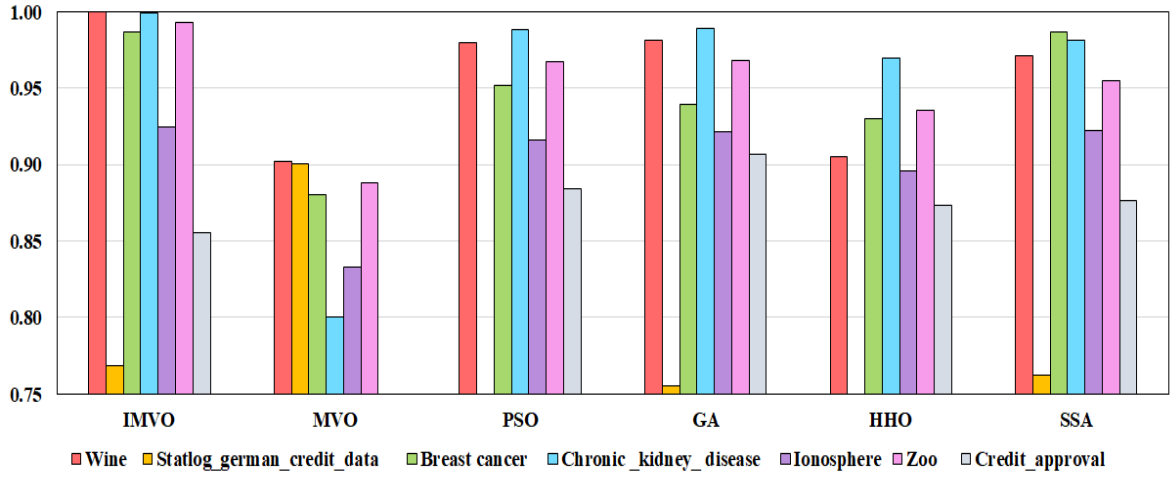
Accuracy of Classification

Table 4 presents the average classification accuracy and standard deviation for each algorithm. The IMVO algorithm consistently achieves the highest average accuracy across all datasets, with these values highlighted in bold in Table 4 and illustrated in Fig. 5. Although PSO and MVO slightly surpass IMVO on the Credit Approval and German Credit datasets, respectively, it is important to note that IMVO selects fewer features for these datasets, showcasing its efficiency in feature selection.

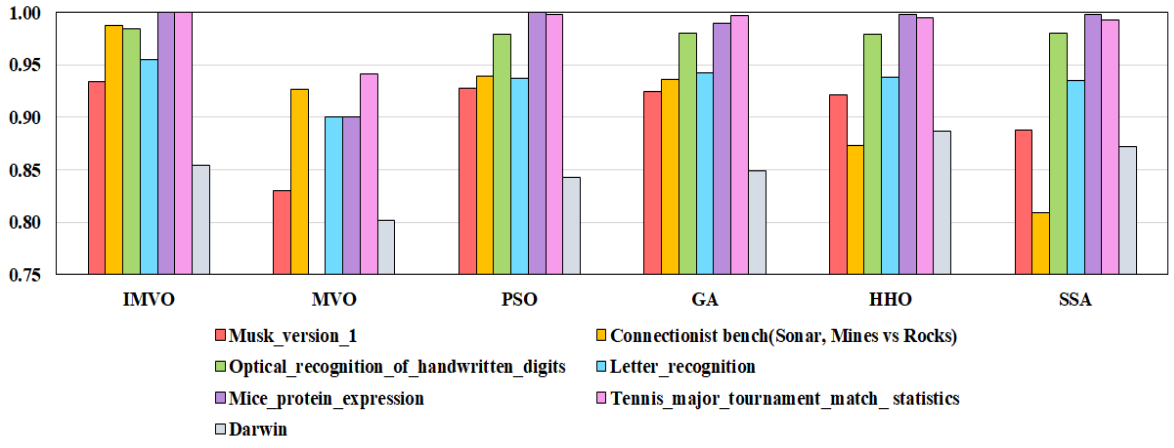
Table 4. Comparison of Classification Accuracy and Its Standard Deviation Across Datasets Over 30 Runs, Indicating Superior Results

Accuracy	IMVO		PSO		MVO		HHO		SSA		GA	
Dataset	**AVG**	**STD**	**AVG**	**STD**	**AVG**	**STD**	**AVG**	**STD**	**AVG**	**STD**	**AVG**	**STD**
Breast cancer	**0.9899**	**0.0044**	0.9514	0.0192	0.8800	0.0844	0.9298	0.0083	0.9871	0.0702	0.9395	0.0073
Musk 1	**0.9339**	0.0184	0.9272	**0.0130**	0.8300	0.0194	0.9216	0.0396	0.8881	0.0420	0.9240	0.0206
Wine	**1.0000**	**0.0000**	0.9796	0.0130	0.9018	0.0123	0.9056	0.0077	0.9711	0.0943	0.9814	0.0087
Kidney disease	**0.9994**	**0.0027**	0.9883	0.0041	0.8000	0.0327	0.9694	0.0171	0.9816	0.0306	0.9888	0.0039
Sonar	**0.9869**	0.0140	0.9396	0.0185	0.9268	0.0250	0.8730	0.0326	0.8095	0.0847	0.9365	**0.0130**
Zoo	**0.99**	**0.01**	0.96	0.02	0.88	0.01	0.93	0.01	0.95	0.05	0.96	0.02

	29	**70**	77	04	78	89	54	52	48	38	79	63
Ionosphere	**0.92 46**	**0.01 50**	0.91 60	0.01 86	0.83 29	0.01 60	0.89 62	0.02 04	0.92 26	0.10 38	0.92 13	0.00 44
German credit data	0.76 88	0.01 68	0.74 40	0.01 39	**0.90 05**	0.01 78	0.74 33	0.01 26	0.76 20	0.23 89	0.75 55	**0.00 31**
Mice protein representation	**1.00 00**	0.00 00	1.00 00	0.00 00	0.90 00	0.05 20	0.99 78	0.00 24	0.99 81	0.00 22	0.99 00	0.00 12
Darwin	**0.89 43**	0.02 04	0.84 27	0.03 88	0.80 14	0.02 98	0.88 67	0.02 98	0.87 16	0.09 62	0.84 90	**0.01 34**
Letter recognition	**0.95 45**	0.01 21	0.93 73	0.00 38	0.90 07	0.01 61	0.93 81	0.00 76	0.93 46	0.06 18	0.94 21	0.00 55
Credit approval	0.85 54	0.00 98	**0.88 40**	0.01 42	0.74 58	0.01 98	0.87 34	0.01 35	0.87 63	0.12 66	0.90 66	0.01 39
Tennis match	**1.00 00**	0.00 00	0.99 76	0.00 33	0.94 15	0.01 60	0.99 50	0.00 28	0.99 22	0.00 49	0.99 64	0.00 17
Handwritten digits	**0.98 48**	0.01 11	0.97 94	**0.00 14**	0.71 29	0.01 21	0.97 86	0.00 44	0.98 03	0.02 11	0.98 04	0.00 37



(a)



(b)

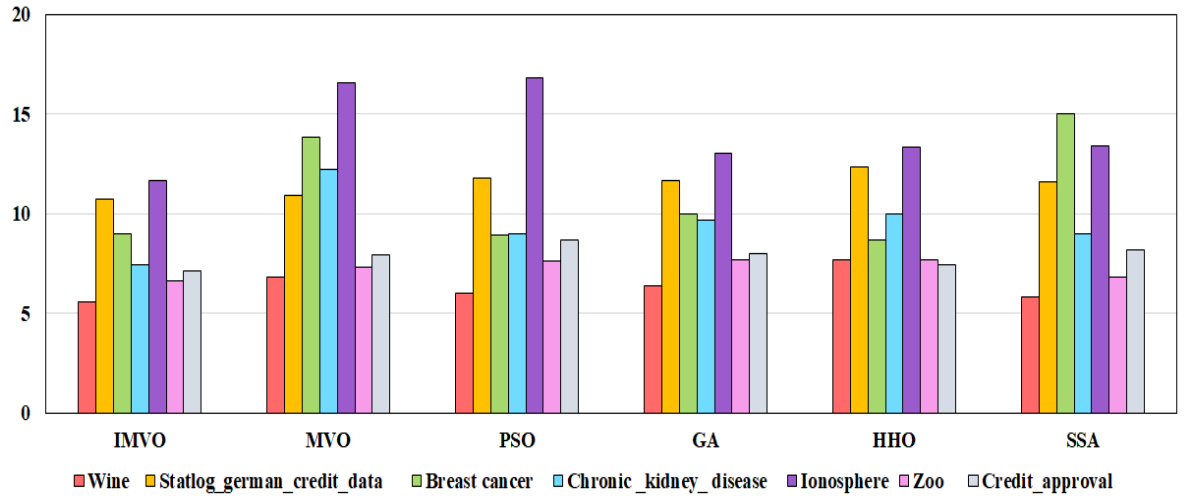
Figure 5. Average Classification Accuracy Across 30 Trials for 14 Benchmark Datasets

Count of Selected Features

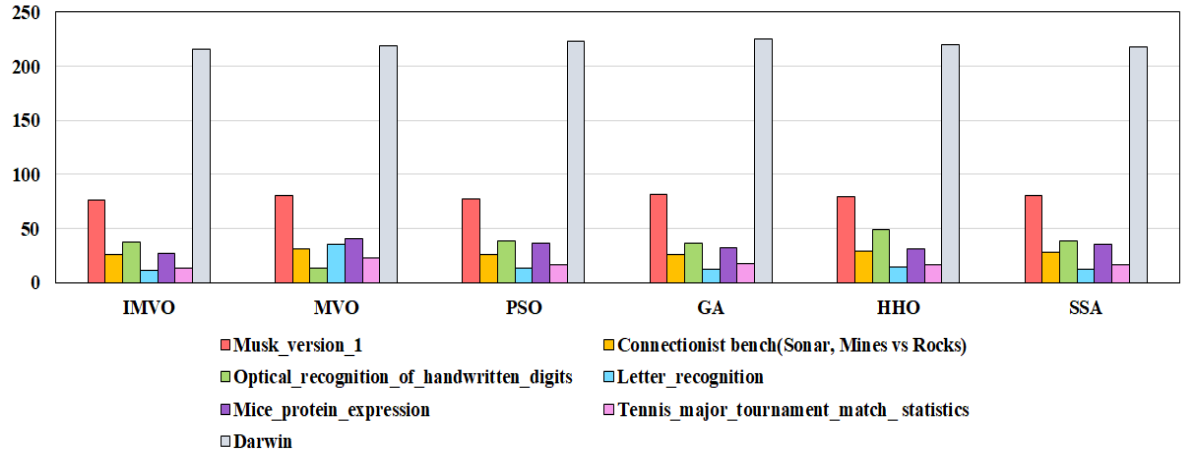
Table 5 presents the average number of features selected by each algorithm. IMVO demonstrates superior performance in this metric, consistently selecting the fewest features in 13 out of 14 datasets. These results are marked in bold in Table 5 and visualized in Fig. 6. The only exception is the Breast Cancer dataset, where GA selects fewer features but at the cost of lower accuracy in comparison with IMVO. This underscores IMVO's capability to maintain a balanced performance, optimizing both feature selection and accuracy efficiency.

Table 5. Average Number of Features Selected by Each Algorithm Across All Datasets, Emphasizing Enhanced Performance

Count of selected features	IMVO	PSO	MVO	HHO	SSA	GA
Breast cancer	9.0000	8.9000	13.8333	8.6667	15.0000	**3.8800**
Musk 1	**76.1000**	77.2000	80.5333	79.0000	80.6000	81.6667
Wine	**5.5667**	6.0000	6.8333	7.6667	5.8000	6.3667
Kidney disease	**7.4500**	9.0000	12.2333	10.0000	9.0000	9.6667
Sonar	**25.9000**	26.0000	31.4333	28.6667	27.4000	26.0667
Zoo	**6.6500**	7.6000	7.3000	7.6667	6.8000	7.6667
Ionosphere	**11.6500**	16.8000	16.5333	13.3333	13.4000	13.0000
German credit data	**10.7500**	11.8000	10.9000	12.3333	11.6000	11.6667
Mice protein representation	**26.5323**	36.0000	39.9500	31.3000	35.4000	32.0000
Darwin	**215.4511**	223.0000	218.3500	220.0000	217.6000	224.6667
Letter recognition	**11.5300**	13.5000	35.6000	13.9000	12.4000	12.6667
Tennis match	**13.1200**	16.3333	22.2000	16.7000	16.2000	17.0000
Credit approval	**7.1000**	8.6667	7.9000	7.4000	8.2000	8.0000
Handwritten digits	**37.4500**	38.0000	13.5	48.7000	38.5000	36.3333



(a)



(b)

Figure 6. Comparison of Feature Selection Efficiency of Algorithms Across All Datasets

Fitness Value

Table 6 summarizes the average fitness values for each algorithm, where the fitness value reflects classification error lower values correspond to better performance. IMVO consistently achieves the smallest classification error across datasets of different sizes, highlighting its robustness and superior effectiveness in comparison with other methods.

Table 6. Algorithm Average Fitness Scores (Lower Values Indicate Better Performance)

Fitness value	IMVO	PSO	MVO	HHO	SSA	GA
Breast cancer	**0.0145**	0.0885	0.0354	0.0618	0.0162	0.8165
Musk 1	**0.0234**	0.0737	0.0823	0.0239	0.0254	0.0535
Wine	**0.0112**	0.0412	0.0426	0.0855	0.0118	0.0214
Kidney disease	**0.0141**	0.0157	0.0271	0.0219	0.0168	0.0145
Sonar	**0.0273**	0.0509	0.0495	0.1310	0.1930	0.0665
Zoo	**0.0212**	0.0363	0.0432	0.0676	0.0250	0.0350
Ionosphere	0.0886	0.0885	**0.0822**	0.0972	0.0897	0.0885
German credit data	0.2443	0.2251	0.2532	0.2616	**0.1463**	0.2449
Mice protein representation	**0.0133**	0.0238	0.0400	0.0139	0.0145	0.024
Darwin	**0.1513**	0.1540	0.1586	0.1542	0.1522	0.1793
Letter recognition	0.0620	0.06871	**0.0410**	0.0703	0.0635	0.0632
Tennis match	**0.0111**	0.0142	0.0125	0.0127	0.0182	0.0273
Credit approval	0.1579	0.1629	0.1671	0.1586	0.1585	**0.0811**
Handwritten digits	**0.0308**	0.0382	0.0603	0.0340	0.0319	0.0350

Significance Analysis

To evaluate the statistical significance of performance differences between the proposed IMVO algorithm and other methods, a Wilcoxon signed-rank test was conducted. This test assesses variations in key metrics, including fitness values, the number of selected features, and classification accuracy. The null hypothesis assumes no significant difference in performance between IMVO and the other algorithms, and it is rejected when the p-value falls below 0.05. The results, presented in Table 7, illustrate that IMVO considerably exceeds all comparison algorithms across all three metrics. For accuracy, the p-values against MVO, PSO, GA, HHO, and SSA are 0.0082, 0.0003, 0.0130, 0.0200, and 0.0292, respectively. IMVO's advantage is even more pronounced in feature selection, with p-values of 0.0105, 0.0001, 0.0021, 0.0005, and 0.0002 for the same algorithms. Similarly, for fitness values, IMVO's excellence is verified with p-values of 0.0050, 0.005, 0.0132, 0.0003, and 0.0104. These findings evidently show that IMVO delivers substantial enhancements in feature selection, accuracy, and fitness values compared to GA, MVO, PSO, SSA, and, HHO refuting the null hypothesis at a 5% significance level.

Beyond the reported p-values, further analysis reveals the underlying reasons for IMVO's significant performance gains. The algorithm's advantage is most evident in high-dimensional datasets such as Darwin, Musk 1, and Mice Protein Representation, where its dual-phase local search mechanism effectively maintains population diversity while refining promising solutions, resulting in higher classification accuracy and more compact feature subsets. This performance edge stems from IMVO's ability to avoid premature convergence through periodic mutations of randomly selected solutions, combined with targeted refinement of the best solutions to accelerate exploitation. These complementary mechanisms consistently yield lower fitness values and improved convergence stability. Even in cases where accuracy gains are modest, such as Credit Approval and German credit data, IMVO achieves statistically significant reductions in the number of selected features, demonstrating benefits in terms of model compactness and interpretability. This integrated statistical and mechanistic interpretation highlights not only the robustness of IMVO across diverse datasets but also the practical value of its design in balancing exploration and exploitation for superior optimization outcomes.

Table 7. Wilcoxon Signed-Rank Test Results

Evaluation	p-value_count of selected features	p-value_average accuracy	p-value_fitness value
IMVO compared to MVO	0.0105	0.0082	0.0050
IMVO compared to PSO	0.0001	0.0003	0.005
IMVO compared to HHO	0.0005	0.0200	0.0003
IMVO compared to SSA	0.0002	0.0292	0.0104
IMVO compared to GA	0.0021	0.0130	0.0132

Algorithmic Complexity

The algorithmic complexity of the proposed algorithms, taking into account factors such as the number of solutions Nsol, problem dimensionality D, maximum iterations T, and the cost of the fitness function C, is analyzed as follows:

The complexity of the MVO algorithm varies with context. In optimal scenarios, the complexity is O (T (Nsol 2+ Nsol Dlog Nsol)), while in worst-case scenarios, it reaches O (TNnum 2(1+D)). When the LSA is integrated, the complexity increases to O (T (Nsol 2+ Nsol Dlog Nsol +Nit)) in the best case and O (T (Nsol 2+ Nsol 2D+ Nit)) in the worst case, where Nit represents the number of LSA iterations, a user-defined parameter. The HHO algorithm has a complexity of O (Nnum (T+TD+1)), whereas both PSO and SSA have a complexity of O (T Nsol (D+C)), reflecting alike computational requirements. For the Genetic Algorithm (GA), the complexity ranges from O (Nnum T (D+C+log Nsol)) in the best case to O (Nsol T (D+C+ Nsol)) in the worst case.

In comparison, the extended MVO algorithm (MVO with LSA) exhibits higher computational complexity than standard MVO, HHO, SSA, PSO, and GA due to the addition of LSA, which enhances solution quality and mitigates the risk of being trapped in local optima. Nonetheless, the Wilcoxon test and convergence analysis reveal that the extended MVO consistently outperforms other algorithms in terms of classification accuracy, fitness value, feature selection, and statistical significance.

Managerial Insights

From a managerial perspective, the findings of this study offer several actionable insights for practitioners and decision-makers seeking to optimize data-driven processes. The proposed IMVO algorithm not only advances the technical state of feature selection but also provides practical benefits that can directly impact organizational efficiency, decision quality, and resource allocation. Key managerial implications include:

- **Algorithm Selection for Feature Selection:** IMVO achieves higher classification accuracy while selecting fewer features compared to competing methods, making it an excellent choice for organizations dealing with high-dimensional datasets where interpretability and computational efficiency are critical.
- **Balanced Exploration and Exploitation:** The hybrid approach (IMVO) effectively avoids local optima, delivering more reliable outcomes in real-world optimization tasks where solution quality is essential.
- **Resource Efficiency:** By identifying a minimal subset of features without sacrificing accuracy, IMVO reduces data storage requirements, processing time, and model complexity benefits that are particularly valuable for cost-sensitive or time-critical projects.
- **Cross-Domain Applicability:** The algorithm's validated performance on diverse datasets demonstrates its adaptability across domains such as healthcare, finance, security, and manufacturing, without the need for extensive customization.
- **Decision-Making Support:** The enhanced interpretability resulting from reduced feature sets allows managers and analysts to better identify and understand the key factors influencing outcomes, thereby supporting more informed and strategic decision-making.

Conclusion

This research evaluates the performance of the IMVO algorithm on multiple benchmark datasets from the UCI repository, emphasizing its robustness and reproducibility. Across all experiments, IMVO consistently surpasses alternative methods such as MVO, PSO, GA, HHO, and SSA, achieving superior classification accuracy and lower error rates while utilizing fewer features. This combination of high accuracy and efficient feature selection demonstrates

IMVO's effectiveness and suitability for optimal feature selection tasks.

Statistical analysis using the Wilcoxon signed-rank test further confirms IMVO's advantages, with p-values below 0.05 indicating significant improvements in accuracy, feature reduction, and fitness values. While incorporating the Local Search Algorithm increases computational requirements, the performance gains provided by IMVO justify this additional cost.

For practical deployment in machine learning workflows, future research should explore IMVO's adaptability across various applications, such as environmental monitoring, industrial automation, and smart transportation. Enhancing its computational efficiency will improve scalability for large datasets. Moreover, refining the local search strategies and integrating IMVO with deep learning approaches could further advance its capabilities, reinforcing its utility in feature selection and classification tasks.

Beyond these directions, explainable AI (XAI) methods, such as SHapley Additive exPlanations (SHAP), can be incorporated to provide deeper insights into the contribution of each selected feature and enhance the interpretability of IMVO-driven results. Additionally, integrating IMVO with ensemble learning models (e.g., Random Forests, XGBoost, or stacking architectures) could improve classification robustness by combining the strengths of multiple learners. Another promising research path involves applying IMVO in federated learning environments, enabling privacy-preserving feature selection across distributed data sources without sharing raw data. Finally, extending IMVO to handle streaming and evolving datasets by designing adaptive feature selection strategies would make the algorithm more suitable for real-time applications such as intrusion detection, IoT analytics, and financial forecasting.

Data availability: The datasets are available in UCI repository, [https://archive.ics.uci.edu/].

References

[1] J. R. Vergara and P. A. Estévez, "A review of feature selection methods based on mutual information," Neural Comput. Appl., vol. 24, no. 1, pp. 175–186, Jan. 2014, doi: 10.1007/S00521-013-1368-0/METRICS.

[2] M. DASH and H. LIU, "Feature selection for classification," Intell. Data Anal., vol. 1, no. 1–4, pp. 131–156, Jan. 1997, doi: 10.1016/S1088-467X (97)00008-5.

[3] R. Sihwail, K. Omar, K. A. Z. Ariffin, and M. Tubishat, "Improved Harris Hawks Optimization Using Elite Opposition-Based Learning and Novel Search Mechanism for Feature Selection," IEEE Access, vol. 8, pp. 121127–121145, 2020, doi: 10.1109/ACCESS.2020.3006473.

[4] K. H. Sheikh et al., "EHHM: Electrical harmony based hybrid meta-heuristic for feature selection," IEEE Access, vol. 8, pp. 158125–158141, 2020, doi: 10.1109/ACCESS.2020.3019809.

[5] B. Xue, M. Zhang, W. N. Browne, and X. Yao, "A Survey on Evolutionary Computation Approaches to Feature Selection," IEEE Trans. Evol. Comput., vol. 20, no. 4, pp. 606–626, 2016, doi: 10.1109/TEVC.2015.2504420.

[6] M. Du, K. Wang, Z. Xia, and Y. Zhang, "Differential Privacy Preserving of Training Model in Wireless Big Data with Edge Computing," IEEE Trans. Big Data, vol. 6, no. 2, pp. 283–295, 2018, doi: 10.1109/tbdata.2018.2829886.

[7] F. Peres and M. Castelli, "Combinatorial Optimization Problems and Metaheuristics: Review, Challenges, Design, and Development," Appl. Sci., vol. 11, no. 14, p. 6449, Jul. 2021, doi: 10.3390/app11146449.

[8] D. H. Wolpert and W. G. Macready, "No free lunch theorems for optimization," IEEE Trans. Evol. Comput., vol. 1, no. 1, pp. 67–82, 1997, doi: 10.1109/4235.585893.

[9] Z. Sadeghian, E. Akbari, H. Nematzadeh, and H. Motameni, "A review of feature selection methods based on meta-heuristic algorithms," J. Exp. Theor. Artif. Intell., 2023, doi: 10.1080/0952813X.2023.2183267.

[10] H. Rao et al., "Feature selection based on artificial bee colony and gradient boosting decision tree," Appl. Soft Comput. J., vol. 74, pp. 634–642, 2019, doi: 10.1016/j.asoc.2018.10.036.

[11] Z. M. Elgamal, N. B. M. Yasin, M. Tubishat, M. Alswaitti, and S. Mirjalili, "An improved harris hawks optimization algorithm with simulated annealing for feature selection in the medical field," IEEE Access, vol. 8, pp. 186638–186652, 2020, doi: 10.1109/ACCESS.2020.3029728.

[12] M. Tubishat, N. Idris, L. Shuib, M. A. M. Abushariah, and S. Mirjalili, "Improved Salp Swarm Algorithm based on opposition based learning and novel local search algorithm for feature selection," Expert Syst. Appl., vol. 145, 2020, doi: 10.1016/j.eswa.2019.113122.

[13] N. Neggaz, E. H. Houssein, and K. Hussain, “An efficient henry gas solubility optimization for feature selection,” Expert Syst. Appl., vol. 152, 2020, doi: 10.1016/j.eswa.2020.113364.
[14] P. Hu, J. S. Pan, and S. C. Chu, “Improved Binary Grey Wolf Optimizer and Its application for feature selection,” Knowledge-Based Syst., vol. 195, 2020, doi: 10.1016/j.knosys.2020.105746.
[15] F. Kılıç, Y. Kaya, and S. Yildirim, “A novel multi population based particle swarm optimization for feature selection,” Knowledge-Based Syst., vol. 219, 2021, doi: 10.1016/j.knosys.2021.106894.
[16] H. Alazzam, A. Sharieh, and K. E. Sabri, “A feature selection algorithm for intrusion detection system based on Pigeon Inspired Optimizer,” Expert Syst. Appl., vol. 148, 2020, doi: 10.1016/j.eswa.2020.113249.
[17] M. Abdel-Basset, G. Manogaran, D. El-Shahat, and S. Mirjalili, “A hybrid whale optimization algorithm based on local search strategy for the permutation flow shop scheduling problem,” Futur. Gener. Comput. Syst., vol. 85, pp. 129–145, 2018, doi: 10.1016/j.future.2018.03.020.
[18] M. Tubishat, M. Alswaitti, S. Mirjalili, M. A. Al-Garadi, M. T. Alrashdan, and T. A. Rana, “Dynamic butterfly optimization algorithm for feature selection,” IEEE Access, vol. 8, pp. 194303–194314, 2020, doi: 10.1109/ACCESS.2020.3033757.
[19] M. D. Toksari, “A hybrid algorithm of Ant Colony Optimization (ACO) and Iterated Local Search (ILS) for estimating electricity domestic consumption: Case of Turkey,” Int. J. Electr. Power Energy Syst., vol. 78, pp. 776–782, 2016, doi: 10.1016/j.ijepes.2015.12.032.
[20] C. Yan, J. Ma, H. Luo, and A. Patel, “Hybrid binary Coral Reefs Optimization algorithm with Simulated Annealing for Feature Selection in high-dimensional biomedical datasets,” Chemom. Intell. Lab. Syst., vol. 184, pp. 102–111, 2019, doi: 10.1016/j.chemolab.2018.11.010.
[21] M. Shehab, A. T. Khader, M. A. Al-Betar, and L. M. Abualigah, “Hybridizing cuckoo search algorithm with hill climbing for numerical optimization problems,” ICIT 2017 - 8th Int. Conf. Inf. Technol. Proc., pp. 36–43, 2017, doi: 10.1109/ICITECH.2017.8079912.
[22] S. Mirjalili, S. M. Mirjalili, and A. Hatamlou, “Multi-Verse Optimizer: a nature-inspired algorithm for global optimization,” Neural Comput. Appl., vol. 27, no. 2, pp. 495–513, 2016, doi: 10.1007/s00521-015-1870-7.
[23] M. Tubishat, Z. Rawshdeh, H. Jarrah, Z. M. Elgamal, A. Elnagar, and M. T. Alrashdan, “Dynamic generalized normal distribution optimization for feature selection,” Neural Comput. Appl., vol. 34, no. 20, pp. 17355–17370, 2022, doi: 10.1007/s00521-022-07398-9.
[24] M. M. Mafarja and S. Mirjalili, “Hybrid Whale Optimization Algorithm with simulated annealing for feature selection,” Neurocomputing, vol. 260, pp. 302–312, 2017, doi: 10.1016/j.neucom.2017.04.053.
[25] Q. Al-Tashi, S. J. Abdul Kadir, H. M. Rais, S. Mirjalili, and H. Alhussian, “Binary Optimization Using Hybrid Grey Wolf Optimization for Feature Selection,” IEEE Access, vol. 7, pp. 39496–39508, 2019, doi: 10.1109/ACCESS.2019.2906757.
[26] E. Emary, H. M. Zawbaa, and A. E. Hassanien, “Binary ant lion approaches for feature selection,” Neurocomputing, vol. 213, pp. 54–65, 2016, doi: 10.1016/j.neucom.2016.03.101.
[27] S. B. Imandoust and M. Bolandraftar, “Application of K-Nearest Neighbor (KNN) Approach for Predicting Economic Events: Theoretical Background,” Int. J. Eng. Res. Appl., vol. 3, no. 5, pp. 605–610, 2013, Accessed: Jul. 19, 2024. [Online]. Available: www.ijera.com
[28] L. Wang, L. Khan, and B. Thuraisingham, “An effective evidence theory based K-nearest neighbor (KNN) classification,” Proc. - 2008 IEEE/WIC/ACM Int. Conf. Web Intell. WI 2008, pp. 797–801, 2008, doi: 10.1109/WIIAT.2008.411.
[29] K. Hussain, N. Neggaz, W. Zhu, and E. H. Houssein, “An efficient hybrid sine-cosine Harris hawks optimization for low and high-dimensional feature selection,” Expert Syst. Appl., vol. 176, p. 114778, Aug. 2021, doi: 10.1016/J.ESWA.2021.114778.
[30] J. Too and S. Mirjalili, “A Hyper Learning Binary Dragonfly Algorithm for Feature Selection: A COVID-19 Case Study,” Knowledge-Based Syst., vol. 212, p. 106553, Jan. 2021, doi: 10.1016/J.KNOSYS.2020.106553.
[31] I. Aljarah, M. Mafarja, A. A. Heidari, H. Faris, Y. Zhang, and S. Mirjalili, “Asynchronous accelerating multi-leader salp chains for feature selection,” Appl. Soft Comput., vol. 71, pp. 964–979, Oct. 2018, doi: 10.1016/J.ASOC.2018.07.040.
[32] M. Mafarja et al., “Evolutionary Population Dynamics and Grasshopper Optimization approaches for feature selection problems,” Knowledge-Based Syst., vol. 145, pp. 25–45, Apr. 2018, doi: 10.1016/J.KNOSYS.2017.12.037.
[33] M. M. Mafarja and S. Mirjalili, “Hybrid Whale Optimization Algorithm with simulated annealing for feature selection,” Neurocomputing, vol. 260, pp. 302–312, Oct. 2017, doi: 10.1016/J.NEUCOM.2017.04.053.
[34] M. Tubishat, Z. Rawshdeh, H. Jarrah, Z. M. Elgamal, A. Elnagar, and M. T. Alrashdan, “Dynamic generalized normal distribution optimization for feature selection,” Neural Comput. Appl., vol. 34, no. 20, pp. 17355–17370, Oct. 2022, doi: 10.1007/s00521-022-07398-9.
[35] M. Tubishat et al., “Dynamic Salp swarm algorithm for feature selection,” Expert Syst. Appl., vol. 164, p.

113873, Feb. 2021, doi: 10.1016/j.eswa.2020.113873.
[36] A. E. Hegazy, M. A. Makhlouf, and G. S. El-Tawel, "Improved salp swarm algorithm for feature selection," J. King Saud Univ. - Comput. Inf. Sci., vol. 32, no. 3, pp. 335–344, Mar. 2020, doi: 10.1016/j.jksuci.2018.06.003.
[37] A. E. Hegazy, M. A. Makhlouf, and G. S. El-Tawel, "Feature Selection Using Chaotic Salp Swarm Algorithm for Data Classification," Arab. J. Sci. Eng., vol. 44, no. 4, pp. 3801–3816, Apr. 2019, doi: 10.1007/s13369-018-3680-6.
[38] S. Arora and P. Anand, "Binary butterfly optimization approaches for feature selection," Expert Syst. Appl., vol. 116, pp. 147–160, Feb. 2019, doi: 10.1016/j.eswa.2018.08.051.
[39] T. Thaher, A. A. Heidari, M. Mafarja, J. S. Dong, and S. Mirjalili, "Binary Harris Hawks Optimizer for High-Dimensional, Low Sample Size Feature Selection," 2020, pp. 251–272. doi: 10.1007/978-981-32-9990-0_12.
[40] E. BAŞ and E. ÜLKER, "An efficient binary social spider algorithm for feature selection problem," Expert Syst. Appl., vol. 146, p. 113185, May 2020, doi: 10.1016/j.eswa.2020.113185.