Journal of King Saud University – Computer and Information Sciences 35 (2023) 101704

HOSTED BY

ELSEVIER

Contents lists available at ScienceDirect

Journal of King Saud University – Computer and Information Sciences

journal homepage: www.sciencedirect.com

A novel improved lemurs optimization algorithm for feature selection problems

Ra'ed M. Al-Khatib [a], Nour Elhuda A. Al-qudah [a], Mahmoud S. Jawarneh [b], Asef Al-Khateeb [c]

[a] Department of Computer Sciences, Yarmouk University, Irbid 21163, Jordan
[b] Faculty of Information Technology, Applied Science Private University, Amman, Jordan
[c] MIS Department, College of Business Administration, King Faisal University, Al Ahsa 31982, Saudi Arabia

A R T I C L E I N F O

Article history:
Received 25 April 2023
Accepted 3 August 2023
Available online 12 August 2023

Keywords:
Artificial Intelligence (AI)
Lemurs Optimization (LO)
Feature Selection (FS)
U-shaped transfer function
Local Search Algorithm (LSA)

A B S T R A C T

The irrelevant and repeated features in high-dimensional datasets can negatively affect the final performance and accuracy of classification-based models. Therefore, feature selection (FS) techniques can be used to determine the most optimal relevant features. In this paper, we fuse a new enhanced model from Lemurs Optimization (LO) algorithm, called Enhanced Lemurs Optimization (ELO). We combine Opposition Based Learning (OBL) and Local Search Algorithm (LSA) to address exploration and exploitation challenges, respectively. Our proposed ELO algorithm incorporates U-shaped and Sigmoid transfer functions during the position update step, leading to improved accuracy and convergence. These new deployments based on the U-shaped and Sigmoid transfer functions are called ELO-U and ELO-S algorithms, respectively. The performance of all three new versions of our proposed optimization algorithms (ELO, ELO-U, and ELO-S) has been evaluated using 21 UCI datasets in different fields and sizes. Moreover, their results are also compared to other competitive algorithms. The evaluation process included several measurements such as fitness value, an average of selected features, and average accuracy. Experimental results demonstrate that our proposed ELO-U algorithm achieves the best average accuracy of 91.03%. Statistical analysis using Friedman and Wilcoxon tests confirms the superiority of ELO-U over other competitors.

1. Introduction

In search space, finding the best and most suitable subset with a large number of decision variables after evaluating all possible subsets is time-consuming. Therefore, adopting any expansive and spacious search algorithm for FS is impractical. Meanwhile, FS can presume larwill generate 2^d possible subsets to be evaluated as the expensive approach, which is acknowledged as an NP-hard problem (Mafarja et al., 2018). Many methodologies that depend on randomness have been recently presented to crack the FS problem. These randomness-based approaches can be practical and appropriate for saving time and resources (Gao et al., 2020; Agrawal et al., 2021). Applying FS in preprocessing step of learning algorithms can lead to simplifying learning outcomes, improving learning accuracy, and reducing learning time (Cai et al., 2018). The productivity and efficiency of meta-heuristics algorithms have been confirmed in cracking complex and large-scale problems in various applications. There are four primary types of meta-heuristic models: memory-based (e.g., tabu search), physics-based (e.g., gravitational search algorithm), evolutionary-based (e.g., differential evolution), and swarm-based like SSA algorithm (Hernandez-Castro et al., 2009; Xue et al., 2015). When employing these algorithms, two common conflicting issues are typically assessed: i) exploiting the best solutions found and maintaining diversity; ii) exploring the search space to discover better solutions (Hernandez-Castro et al., 2009). Lately, a good evolutionary algorithm has been presented by (Abasi et al., 2022), named the Lemurs Optimizer (LO) algorithm, which is inspired by lemurs' locomotor manners. The LO uses two movements, 'leap up' and 'dance hup', to guide the search for optimal solutions. These behaviours are

E-mail addresses: raed.m.alkhatib@yu.edu.jo (R.M. Al-Khatib), nourelhudaalqudah1985@gmail.com (N.E.A. Al-qudah), m_jawarneh@asu.edu.jo (M.S. Jawarneh), amaalkhateeb@kfu.edu.sa (A. Al-Khateeb)
Peer review under responsibility of King Saud University.

ELSEVIER Production and hosting by Elsevier

https://doi.org/10.1016/j.jksuci.2023.101704
1319-1578/

adapted in terms of optimization application for tackling FS problems, which made LO algorithm a promising approach.

In a brief recap, the core contributions of our research can be outlined as follows:

- Optimize Lemurs algorithm to tackle FS problem.
- Two approaches have been suggested to improve the original LO algorithm. These aim to address the problems of low diversity and getting stuck in local optima that the Lemurs face. The first approach involves using OBL during population phase to enhance the initialization stage. Then, the second approach compares two different types of Transfer Functions (TF), such as U-shape TF and S-shape TF for binarise continuous values in updated lemur positions. In addition to that LSA is deployed to improve the exploitation stage, which is taken and adapted from (Tubishat et al., 2020).
- Comparative evaluation for LO algorithm, and our proposed ELO algorithm using accuracy, F1_scores, value of fitness (mean, Std), recall, and optimal volume of selected feature parts that generate the highest performance in accuracy and F1_scores.
- The algorithms' performance is evaluated among 21 datasets from UCI repository in different sizes.
- The best enhancements for the original version of LO algorithm and the two versions of ELO will be compared against several algorithms in the literature (Hegazy et al., 2020), such as Ant Lion Optimization (ALO), Grey wolf optimization (GWO), Particle Swarm Optimization (PSO), Salp Swarm Algorithm (SSA), Genetic Algorithm (GA), and Improved Salp Swarm Algorithm (ISSA) algorithm.

The rest is structured into several parts. Section 2 discusses the recent optimization of related-work algorithms that are adapted as main wrapper methods to address FS problem. Section 3 presents a detailed explanation of how we implement the Lemurs Optimizer (LO) algorithm, then, we introduce and deploy a novel approach called LSA to address the exploitation deficiency in the original LO algorithm. Additionally, we utilize the power of OBL in the LO initialization process to improve exploration and diversity. These new improvements by deploying LSA and OBL with the original LO algorithm, are called ELO algorithm. We also present a binary version of ELO, which is developed by using a novel U-shaped TF and a Sigmoid TF, resulting in two variations of ELO: ELO-U and ELO-S, respectively. In Section 4, we exhibit the experimental findings of LO compared to our proposed ELO-U and ELO-S using evaluation matrices. Furthermore, we compare the best binary enhancement version, which is the ELO-U algorithm using accuracy matrices and *p-value*. Finally, in Section 5, the paper ends by emphasising the proposed algorithms' efficacy in tackling the FS problem.

2. Related work

Numerous algorithms in literary works have been fine-tuned to address the FS problem that aims to obtain the most accurate or nearly the best solution for decision-making problems. Such algorithms are considered derivative-free techniques, often involving a trial-and-error method to find satisfactory solutions. They also offer flexibility, clarity, simplicity, and the capability to circumvent the problem of local optima. These algorithms can be easily adapted to specific problems, and they can be modified as needed. One of the strengths of search algorithms is to overcome the local optima, which is the tendency for the algorithm to converge the suboptimal solution before the global optima are reached. Metaheuristic algorithms involve a trade-off between exploitation and exploration as two main fundamental aspects (Olorunda and Engelbrecht, 2008; Mirjalili et al., 2017). In this related work, we demonstrate and discuss the latest evaluation-based algorithms adapted from natural inspiration, such as GWO (Mirjalili et al., 2014), GA (Michalewicz, 1994), PSO, and SSA (Mirjalili et al., 2017). Recent studies have presented some enhancements to these Metaheuristic algorithms to improve their performance (Al-Khatib et al., 2019; Doush et al., 2018). For example, enhancements to SSA include ISSA (Hegazy et al., 2020), which introduces ω similar to PSO, and DSSA (Tubishat et al., 2021), which utilizes the singer's chaotic map and LSA. Also, ISSA was further enhanced in (Tubishat et al., 2020), by incorporating OBL and LSA to increase diversity and improve exploration. The results were promising, as the enhanced SSA showed exceptional performance compared to other search algorithms on UCI datasets. These enhancements have demonstrated high accuracy rates and minimal feature selection. Similarly, improvements to PSO have been achieved through hybrid approaches like HPSO-LS proposed in (Moradi and Gholampour, 2016), which incorporated a local search strategy for detecting minimally correlated feature subsets. Experimental results showed higher classification accuracy compared to other techniques. In (Song et al., 2022), a combined algorithm called SS-PSO, which is Surrogate Sample-Assisted Particle Swarm Optimization algorithm that aims to address the FS problem by reducing computational costs while maintaining high prediction accuracy. It divided the sample set into smaller subsets by employing a non-repetitive uniform sampling technique, and then each subset will be treated alone as a surrogate unit. After that, they introduced a collaborative feature clustering method to reduce computational costs and search space. To minimize particle evaluation costs, all samples with surrogate units are replaced using PSO based on ensemble function of surrogate-assisted integer. The final results of this hybrid SS-PSO model achieved good feature subsets at a reduced computational cost compared to other algorithms, which makes SS-PSO a powerful algorithm for solving high-dimensional feature selection. BGWOPSO is another binary algorithm that was derived from the combination of GWO and PSO by (Hu et al., 2020). Using the Euclidean separation metric, it also employed a wrapper-based approach with K-nearest neighbours classifier. BGWOPSO was evaluated based on 18 UCI benchmark datasets, demonstrating its superiority over other binary algorithms in accuracy, optimal feature selection, and computational time.

In the field of medical diagnosis, FS algorithms have made significant advancements. SSA-FS was the first method proposed by optimizing SSA to solve the problem of FS and tested on natural and synthetic biomedical datasets in (Ibrahim et al., 2017; Barhoush et al., 2023). A combination of Ant Lion Optimization (ALO) with Butterfly Optimization Algorithm (BOA) was presented in (Thawkar et al., 2021; Alkhateeb et al., 2020) to select optimal feature subsets for breast cancer diagnosis. This proposed technique outperformed individual BOA and ALO methods, showcasing its potential for clinical applications. Furthermore, the Modified ALO (MALO) algorithm has been introduced in (Gupta et al., 2020) in the context of thyroid disease diagnosis, which significantly improved accuracy and reduced the computational time by extracting relevant features. Experimental results have demonstrated the good performance of MALO compared against other algorithms. In a different approach, (Mafarja and Mirjalili, 2017) presented a wrapper-based algorithm that combined the Whale Optimization Algorithm (WOA) with the Simulated Annealing (SA) algorithm. This combined algorithm improved the exploitation abilities of WOA in feature selection, by using 18 UCI datasets in evaluation tests. Overall, the reviewed literature highlights effectiveness of developing optimization algorithms, which are reducing computational complexity, and enhancing classification accuracy in various domains, including medical diagnosis. These advancements contribute to more efficient and accurate diagnostic models.

Moreover, the limitation of Sine Cosine Algorithm (SCA) in exploration and diversity phase, was solved by deploying OBL with SCA in (Gupta and Deep, 2019). U-shape Transfer Function was introduced in a research paper by Mirjalili (Mirjalili et al., 2020), and then it was employed in various studies to tackle NP problems like FS problem and 0/1 Knapsack problem. For instance, researchers in repulsive binary algorithms utilized U-shape TF in a project outlined in (Ervural and Hakli, 2023) to address the problem of 0/1 Knapsack. Similarly, in (Chen et al., 2023) the authors combined V-shape and U-shape TF to reduce dataset dimensionality in binary PSO. Additionally, the U-shape TF played a crucial role in discretizing the EA with SA in DEOSA, as explained in (Guha et al., 2023). Using U-shape TF led to a higher convergence rate in the DEOSA analysis of repository datasets, where further information in (Pan et al., 2022).

For those seeking a thorough grasp of the enhanced techniques, as well as the pros and cons of evolutionary algorithms for feature selection, there are valuable reviews accessible on (Zhang et al., 2019; Agrawal et al., 2021; Nguyen et al., 2020). Summary of some methods in Table 1 describes recent related works that addressed FS problem.

Our research attempts to address the gap by devising another optimization algorithm for FS problems, which incorporates the simulation of lemur's social behaviour in nature, particularly the leap-up behaviour and dance-hup strategy.

3. Research methodology

This section will first explain the inspiration and mathematical model behind LO algorithm. Then, we will discuss our proposed enhancement approach by incorporating Opposition Based Learning (OBL) during the initialization phase. Additionally, we will introduce the Local Search Algorithm (LSA), which enhances the exploitation stage within the context of the LO algorithm. Finally, we will describe our newly developed model based on deploying U-shape and Sigmoid transfer functions. These functions convert continuous values into binary while updating the lemurs' position.

3.1. Lemurs Optimizer (LO)

This section provides an overview of the inspiration behind LO algorithm, with the mathematical model of LO algorithm.

3.1.1. Inspiration of LO algorithm

Lemurs are a type of prosimian primate, which are distinct from monkeys and apes. These animals are native to Madagascar and Comoro Islands, and there are only a few individuals of each species. Lemurs inhabit various forest environments, including marshlands, dry forests, and rainforests (Jolly, 1966). The Lemurs Optimizer algorithm was firstly developed by (Abasi et al., 2022), which draws inspiration from two primary lemurs behaviors: the leap up behavior, and the dance hup behavior. When committing a leap up, they propel themselves up into air and then land in an upright sitting position on a tree while holding on with both hands and feet (see Fig. 1 (A)). When the distance between trees is excessive, the lemur performs dance-hup by plunging to the ground to jump horizontally using his arms expanded to the side and waving up and down to maintain balance while jumping high (see Fig. 1 (B)) (Powzyk and Mowry, 2007).

In any meta-heuristic optimization algorithm, balancing exploitation and exploration is crucial. The leap-up behavior of lemurs represents the exploration phase of the LO algorithm, which aids in the identification of the optimal lemurs within the search space. The dance-hup behavior aids in exploitation, as lemurs move towards the best nearby location in one direction (Abasi et al., 2022). Our enhancement to the LO algorithm is performed by utilizing OBL in the initialization phase, then using the LSA to enhance exploitation. After that, we deployed U-shape or Sigmoid as novel transfer functions to binary the continuous value in updating lemurs' position phase.

3.1.2. Mathematical LO model

This part introduces mathematical formulas for LO algorithm, the flowchart, and the steps of algorithm. LO is known as one of powerful population-based algorithms so, the lemurs set is expressed in the matrix form. Eq. 1 defines the matrix of the input population for the LO algorithm.

$$X = \begin{bmatrix} l_1^1 & l_1^2 & \cdot & l_1^d \\ l_2^1 & l_2^2 & \cdot & l_2^d \\ \vdots & \vdots & \vdots & \vdots \\ l_n^1 & l_n^2 & \cdot & l_n^d \end{bmatrix} \tag{1}$$

where X indicates the matrix of algorithm set in size $n \times d$. n points to the candidate solutions, and d represents the decision variables. To use the LO for solving an optimization problem like Feature

Table 1
A comparative summary of related works.

Reference	Method	Enhancement	Dataset	Result/Accuracy
(Mafarja and Mirjalili, 2017)	WOA	SA	18 UCI benchmark datasets	Acc = 97.995%
(Tubishat et al., 2020)	SSA	OBL & LSA	18 UCI datasets	Optimal solution.
(Hegazy et al., 2020)	SSA	ω to improve Salp algorithm	23 UCI repository datasets	ω harmonized exploration and exploitation, and accelerated convergence rate.
(Tubishat et al., 2021)	SSA	DSSA using Singer's chaotic map to update Salp's position, & LSA to prevent stuck in local optima	20 UCI datasets & three Hadith datasets	DSSA got better accuracy with selected minimal features.
(Moradi and Gholampour, 2016)	HPSO-LS	hyprid PSO with LSA	13 benchmark classification problems	Achieved higher accuracy with superiority of HPSO-LS.
(Thawkar et al., 2021)	BOA	hybrid BOA with ALO	Breast cancer using mammogram images	Sufficient breast cancer diagnosis that could help in clinical settings
(Gupta and Deep, 2019)	SCA	OBL, self-adaptive component added to SCA search equation	23 datasets with standard IEEE CEC 2014 benchmarks, & 5 engineering	Usefulness for identifying optimal solutions for global optimization problems encountered in real-world applications.
(Gupta et al., 2020)	ALO	MALO	Thyroid disease dataset using image	Acc. 95.94% using RF classifier with a decrease in time and removed 71.5% of inconsequential features
(Ibrahim et al., 2017)	SSA	SSA-FA	Biomedical datasets	Outperformed other methods (PSO, DE) in accuracy with lowest runtime

Fig. 1. Lemurs inspiration behaviour.

selection (FS), the function of the LO algorithm runs into many steps:

Step 1: Define the following Lemurs parameters: N Population, Max_{iter} refers to the maximum number of iterations. d implies the dimensionality of search space over the dataset size. Besides, UB is the upper bound, and LB is the lower bound.

Step 2: Generate X decision variable in i^{th} solution based on Eq. 2:

$$X_i^j = (LB + (UB_j - LB_j)) \times r \tag{2}$$

where r refers to the uniform random number $\in [0, 1]$.

Step 3: Inside the loop for each iteration, calculate the Free Risk Rate (FRR) which is the coefficient of LO algorithm based on Eq. 3:

$$FRR = HRR - t \times ((HRR - LRR)/Max_{iter} \tag{3}$$

where t refers to the present iterations' number. Max_{iter} represents iteration-size. Additionally, Eq. 3 utilizes High Risk Rate (HRR) and Low Risk Rate (LRR) as two predefined and constant values.

Step 4: Calculate the fitness value for each x_i^j, as expressed in Eq. 4:

$$Fit(x_i^j) = \alpha \times (1 - Acc) + \beta \times (s/S) \tag{4}$$

where the fitness value is denoted by $Fit(x_i^j)$, small s refers to the total of selected features, S is the maximum selected feature, and Acc is the accuracy of each subset which is extracted by K-Nearest Neighbors (KNN) classification function to evaluate the selected subset in each iteration.

Step 5: To improve the fitness value of lemurs, We categorize them into two different processes. Firstly, we identify the best near lemurs (bnl), which means selecting the solutions with the lowest fitness value. Based on the FS objectives, bnl will provide the best features for the current iteration. Secondly, we select the global best lemur (gbl) from the entire population, representing the overall best solution.

Step 6: Setting the value of r_1 which is a random number $\in [0, 1]$, and comparing it with FRR. Then, update the position for each lemur away from the risk-based according to Eq. 5.

$$X_i^j = \begin{cases} x(i,j) + |(x(i,j) - x(bnl,j)| \times (r_3 - 0.5) \times 2; & r_1 < FRR \\ x(i,j) + |(x(i,j) - x(gbl,j)| \times (r_3 - 0.5) \times 2; & r_1 > FRR \end{cases} \tag{5}$$

where r_1 is the random number $\in [0, 1]$. The current i^{th} lemur of N^{th} population is $x(i,j)$, which is the candidate solution in the j^{th} decision variable. bnl refers to the best near lemurs, as the present solution in this iteration. gbl implies the global best lemurs for the whole population all over the iterations. Fig. 2 represents the flowchart for LO algorithm.

Algorithm 1 presents the LO algorithm that initiates by randomly developing a swarm of lemurs. Then, it attempts to move toward the lemurs with lower fitness values every iteration through dance hup, as a better fitness value. The optimization procedure randomly starts forming a set of lemurs. The FRR value starts near LRR, indicating that the lemur begins to move and shift towards the best nearest one using the action of 'dance hup'. Now, the function of LO performs this dance hup action to reduce the value of FRR towards HRR. Then, it uses the leap up action to imply the lemur towards the best global solution. This action is reiterated until the final stopping criterion is fulfilled. The movement behavior of lemur members using leap up and dance hup is shown in Fig. 3.

Algorithm 1. Main LO algorithm

```
Define the parameter UB, LB, Max_iter, Dimension
  d, LRR, HRR.
Initialize the size of Lemurs population N.
Generate population randomly.                              ▷ Eq. 2
t = 1
while t < Max_iter do
   Calculate coefficient FRR                               ▷ Eq. 3
   Calculate Fit for each candidate Lemurs
   Sort lemurs candidates
   Update global best candidate lemur gbl
   Update bnl
   for i = 1 to N do
      Set r_1 ← rand[0, 1]
      if r_1 < FRR then
         Update each decision variable using dance hup    ▷ Eq. 5
      else
         Update each decision variable using leap up      ▷ Eq. 5
      end if
   end for
end while
```

3.2. Opposition Based Learning (OBL)

OBL is a popular approach that has been used in many Machine Learning (ML) and computing algorithms (Tizhoosh, 2005; Alaiad et al., 2023), due to its ability to increase the convergence process and improve the effectiveness of metaheuristic outcomes (Rahnamayan et al., 2007). OBL is often used in optimization problems with an ample search space, where the initial location starts

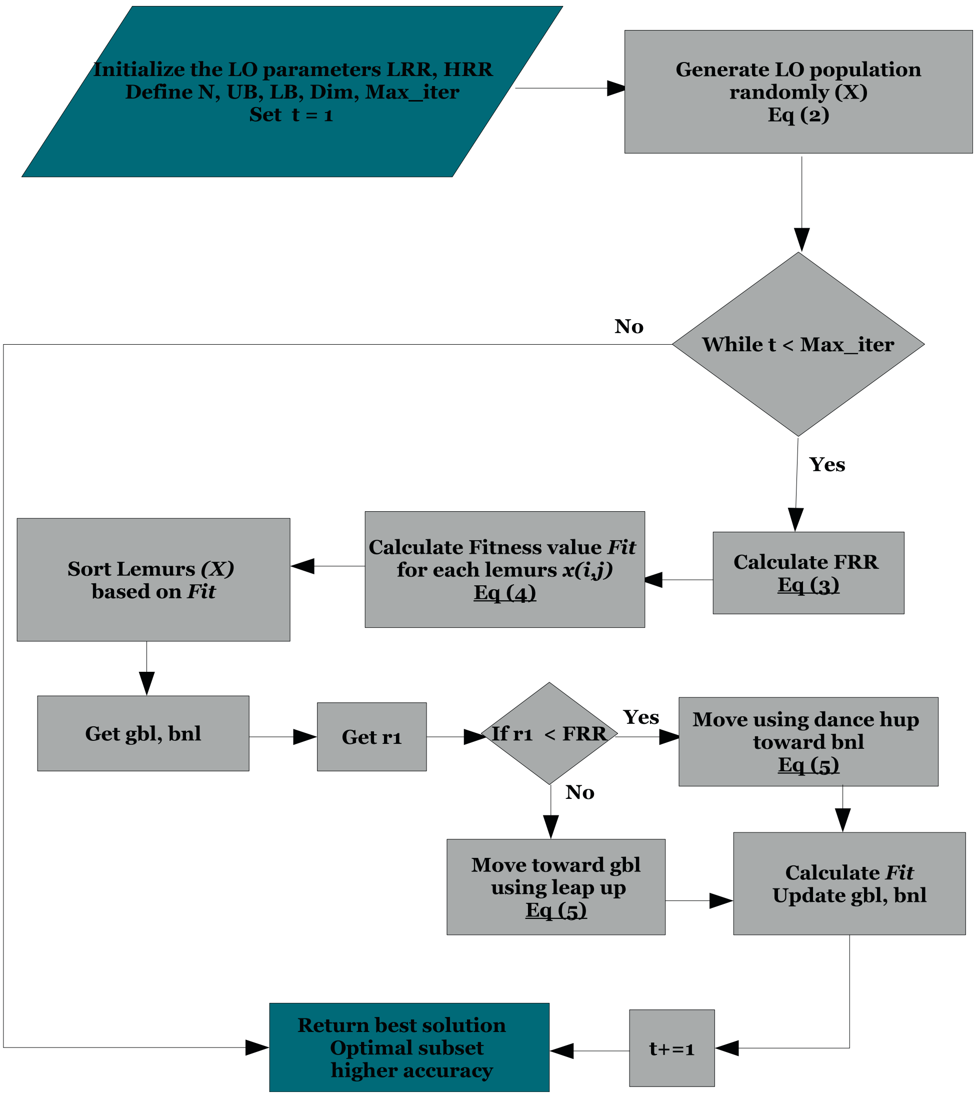

Fig. 2. Flowchart of LO algorithm.

randomly and can be far from the optimal solution. Determining the best location without any prior knowledge is challenging. OBL aims to create adversarial relationships between candidate solutions X and their opposites, denoted as X^o. By considering the opposite of candidate solutions, OBL helps explore the search space more effectively, which leads to obtain better solutions. X^o represents the set of decision variables as in Eq. 6.

$$X^o =< x_1^o, x_2^o, x_3^o, \ldots x_i^o > \tag{6}$$

Then, each decision variable x_i^o in X^o calculated based on Eq. 7:

$$x_i^o = lb_i + ub_i - x_i \tag{7}$$

where the lower boundary is lb_i, and ub_i is the upper boundary of the opposition candidate.

Algorithm 2. OBL Algorithm

Set candidate solutions X
for *each* x_i **do**
  Calculate Opposition Based Learning x_i^o ▷ Eq. 7
end for
for x_i *and* x_i^o **do**
  Calculate *Fit*
  if $Fit(x_i)$ *better* $Fit(x_i^o)$ **then**
    Replace x_i with x_i^o
  end if
end for

3.3. Local Search Algorithm (LSA)

LSA strategy is an important procedure for resolveing difficult computational optimization issues, because LSA is based on sampling to offer and evaluate many possible solutions until they reach the optimal solution. LSA solves major problems through the so-called (search space). This algorithm reduces memory size, so it is useful for pure optimization problems. This technique has been recently used to improve the exploitation phase in many metaheuristic algorithms such as the SSA algorithm (Tubishat et al., 2020). Algorithm 3 shows the LSA algorithm, which runs in the following steps to improve the metaheuristic-based algorithm:

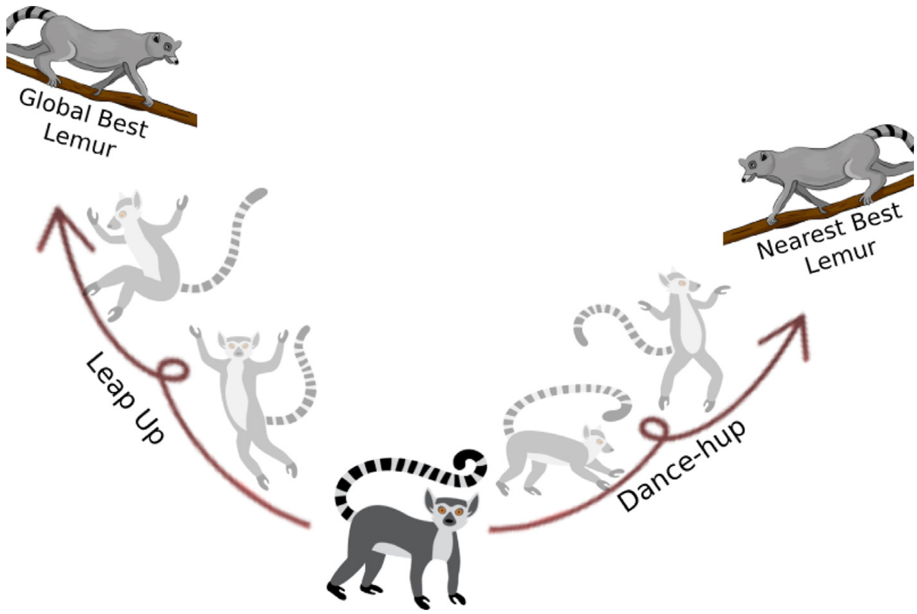

Fig. 3. Lemurs behaviour movement.

Step 1: LSA gets a copy of the X, as copy of decision variables' values called *Temp*.
Step 2: Define Max_{Loc_iter}, *thres*, and set $Lt = 0$ as a current local iteration.
Step 3: Inside loop Max_{Loc_iter} to get the m as random sample of the *Temp*.
Step 4: For each $Temp_i$ in m, update the value of $Temp_i$ line 5 to 11 in Algorithm 3.
Step 5: Apply binary conversion for the new *Temp*.
Step 6: Calculate *Fit* for *Temp*.
Step 7: Compare between $Fit(X)$ and $Fit(Temp)$ and choose the fittest value where the fitness is lowest and replace it with the original candidates.

Algorithm 3. LSA Algorithm

1: $Temp = X.copy$
2: Define $Lt = 0, Max_{Loc\_iter}$, and *thres*
3: **while** $Lt < Max_{Loc\_iter}$ **do**
4: Shuffle a random of *Temp* ▷ Selected Feature
5: **for** Each $Temp_i$
6: **if** $Temp_i > thres$ **then**
7: $Temp_i = Temp_i - thres$
8: **else**
9: $Temp_i = Temp_i + thres$
10: **end if**
11: **end for**
12: Apply Binary Conversion
13: Calculate *Fit* for *Temp*
14: Compare between $Fit(X)$ and $Fit(Temp)$
15: $Lt = Lt + 1$
16: **end while**

3.4. Transfer functions

The features are normalized to be a binary, when representing the selected feature with '1', and '0' for representing the unselected one, as seen in Fig. 4.

In terms of describing values out of continuous values, many research works recently that are started to use the U-shape transfer function for each candidate solution X_i^j based on Eq. 8:

$$TX = \alpha \times (x_i^j)^{\beta} \tag{8}$$

where the control parameters are represented by α and β, Here, the transfer function slope denoted by α, and β implies to the basin width of U-shape transfer function (Mirjalili et al., 2020). α is a predefined uniform variable between [0.5,2]. β also is a predefined uniform variable between [1.5,4], as shown in Fig. 5

Sigmoid transfer function is another powerful transfer function that has been efficiently adapted to solve FS problem. Sigmoid transfer function is illustrated in Fig. 6, which can be deployed for each candidate solution X_i^j based on Eq. 9.

$$TX = \frac{1}{1 + e^{-x_i^j}} \tag{9}$$

Eq. 10 represents the use of *TX* value either in U-shape TF, or in Sigmoid TF for FS problem.

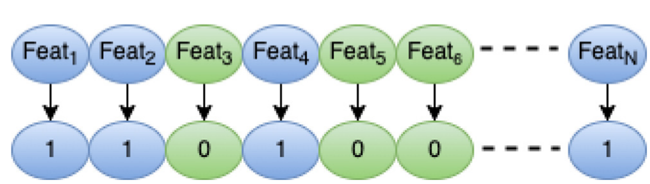

Fig. 4. Binary representing of selected and unselected features.

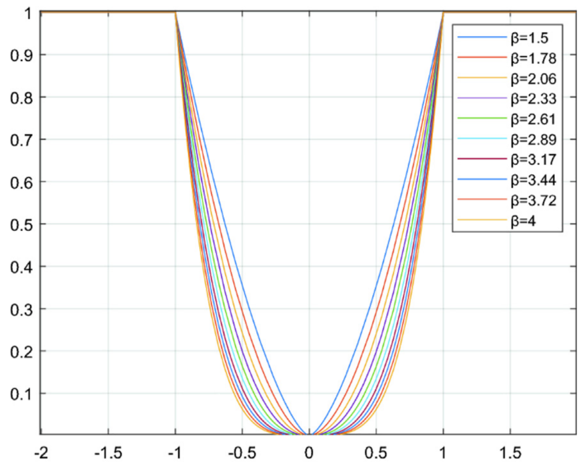

Fig. 5. U-shape transfer function (Mirjalili et al., 2020).

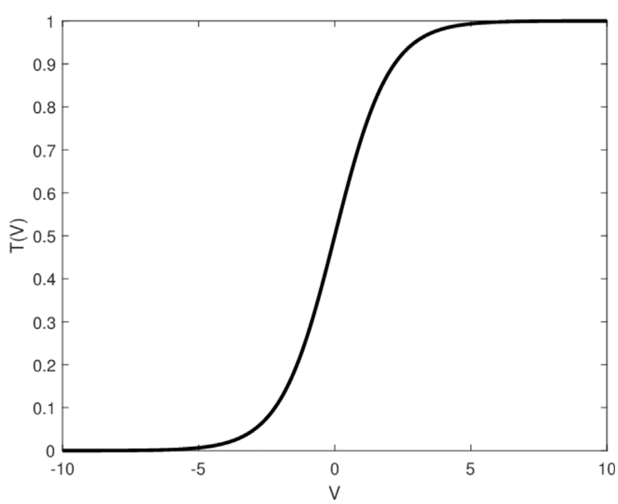

Fig. 6. Sigmoid transfer function (Mafarja et al., 2018).

$$x_i^j = \begin{cases} 1 & , TX > thres \\ 0 & , TX < thres \end{cases} \tag{10}$$

3.5. Enhanced lemurs optimizer

Our proposed ELO algorithm aims to develop the balance between exploration and exploitation in LO algorithm by incorporating OBL in the population initialization stage. To prevent getting trapped in local optima, we employ LSA technique in the exploitation phase of lemurs. Therefore, Our proposed ELO algorithm seeks optimal solutions within the neighbourhood of the search space.

Basically, our proposed ELO algorithm starts by generating an opposite population using OBL, then merges the original and opposite populations. The fitness values for both populations are calculated. Finally, the fittest lemurs enter the optimization loop. After calling LO algorithm, the LSA is applied to find and locate optimal solution by seeking in neighbour search space, which iterated a maximum number of local search iterations.

In our proposed ELO, we deploy one of the novel U-shape TF or the Sigmoid TF to update the lemurs' positions inside the optimization loop, which are called ELO-U and ELO-S, respectively. These both U-shape and Sigmoid transfer functions that are deployed in ELO, will help to distinguish the value of features. Then, they assign '1' to the selected feature, and the unselected feature is assigned to '0'. Fig. 7 represents a chart of ELO algorithm, and the pseudo-code of ELO function is represented in Algorithm 4:

Algorithm 4. Pseudocode of our proposed Enhanced ELO algorithm

Define the parameter ub, lb, Max_{iter}.
Dimension d, LRR, and HRR.
Initialize the size of Lemurs population N.
Generate population randomly ▷ Eq. 2
Set candidate solutions X
Call Algorithm 2 (OBL)
Initial Lemurs Population
while $t < Max_{iter}$ **do**
  Call Lemurs Algorithm 1 (LO)
  Update lemurs position using TF ▷ U-shape or S-shape
  while $Lt < Max_{Loc\_iter}$ **do**
    Call Algorithm 3 (LSA)
  end while
end while
Return an optimal subset of feature with higher accuracy

The complexity of our proposed ELO algorithm is determined by four factors: initialization, fitness function, updating the lemurs' position, and LSA strategy. To estimate the computational complexity of proposed ELO algorithm, consider the following ELO steps:

1. Initialization: $O(N \times dim)$, thus the complexity of initialization process involves OBL and a uniform initialization stage, which is $O(N \times dim)$. Here, N refers to the number of lemurs in the population, and dim represents dimensionality.
2. To update the positions of lemurs and determine the optimal location, the time complexity is $O(Max_{iter} \times N) + O(Max_{iter} \times N \times dim)$. Max_{iter} refers to the maximum number of iterations for LO algorithm, while lemurs' positions are updated in each iteration, which requires $O(N)$ operations. In addition, the optimal location with the highest value of fitness is found among lemurs, but it also requires $O(N)$ operations to be obtained. $O(Max_{iter} \times N)$ is the complexity of the overall process since finding the best location and updating positions is accomplished in each iteration. In addition, updating the positions requires updating every dimension of each lemur's position (dim dimensions), resulting in a complexity of $O(Max_{iter} \times N \times dim)$.
3. Applying LSA Strategy for a maximum number of local search iterations requires an $O(Max_{Loc\_iter})$ operation.

Thus, to determine the overall computational complexity of our proposed ELO algorithm, the approximation can be calculated as in the following equation:

$$O(ELO) = O(N \times dim) + O(Max_{iter} \times N) + O(Max_{iter} \times N \times dim) + O(Max_{Loc\_iter}) \tag{11}$$

3.6. Dataset acquisition

We evaluated the effectiveness of the LO, ELO-U, and ELO-S algorithms using 21 datasets that have been widely used in literature reviews. These 21 datasets are collected and downloaded from UCI repositories. The final obtained comparative results are pro-

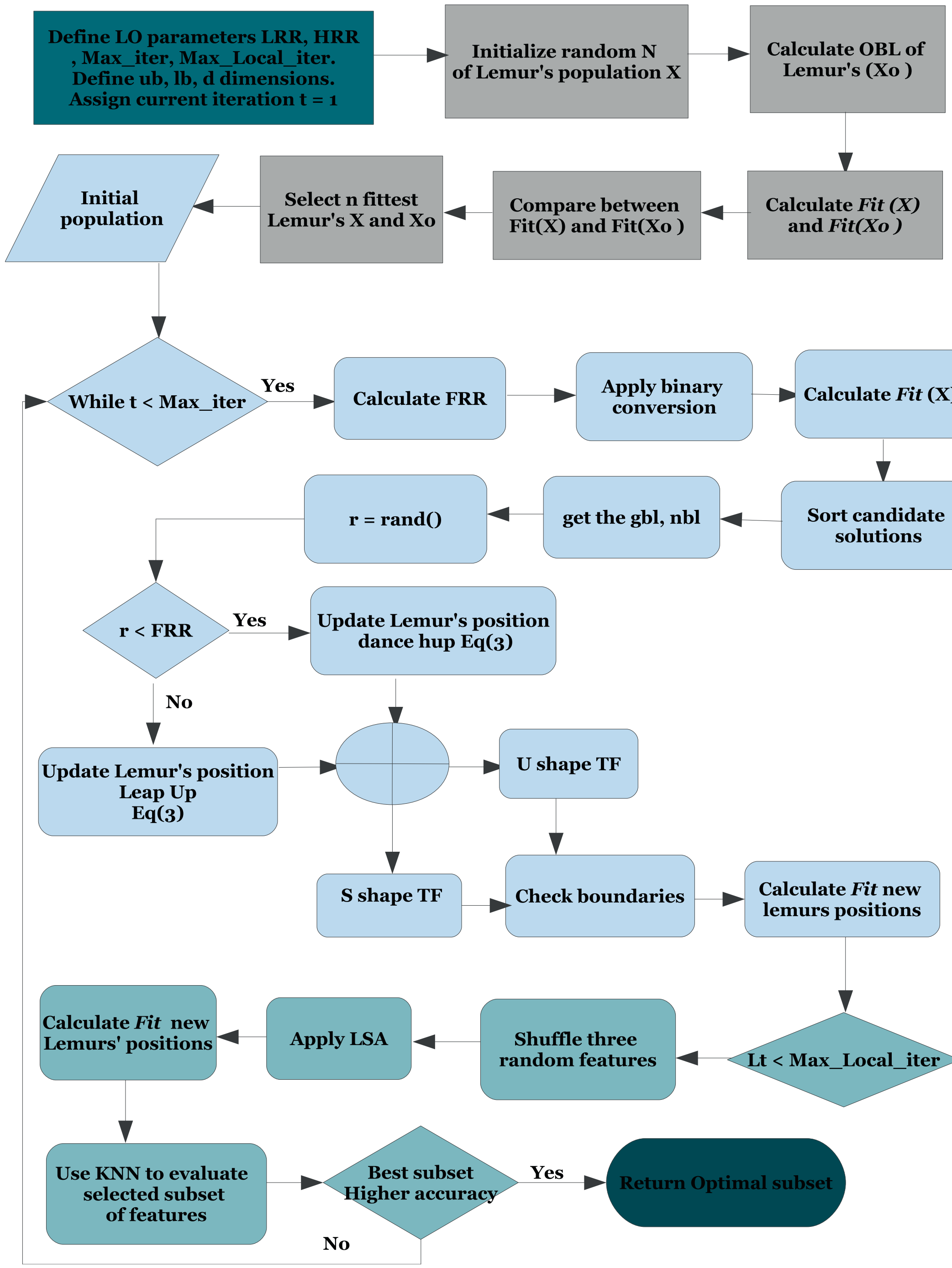

Fig. 7. Framework of proposed ELO algorithm.

cessed according to the experiments in (Hegazy et al., 2020; Al-Khatib et al., 2023). These datasets were used to assess the capabilities of our proposed ELO-U model, and to validate the applicability of deploying the U-shape transfer function within our proposed ELO algorithm. We downloaded the datasets from UCI repository, and processed them using Python machine learning and data science libraries, including Pandas, Numpy, and Sklearn. The datasets are firstly pre-processed by addressing missing or *NaN* values, and labeling the encoding for the object type label. We employed Synthetic Minority Oversampling Technique (SMOTE) to solve the imbalanced datasets, adapted from (Chawla et al., 2002).

The used datasets differ in the number of features, where some have a small size of feature sets, and others have high dimensionality. Table 2 provides the utilized datasets, the total number of used features, detected classes, and instances. Python 3.9 is the used software for implementing all algorithms and comparisons, which utilized a range of libraries such as Pandas, Sklearn, and Seaborn for preprocessing and training models. The library of Matplotlib is also employed to visualize the outcomes of the experiments. All experiments have been done on a MacBook Air with i7 Intel processor 2.2 GHz, 8-GByte RAM, and running on macOS Mojave v. 10.14.

Table 2
Description of used datasets.

Dataset	Feature size	Sample size	# Classes
Wine	14	177	3
Hepatitis	20	155	2
Zoo	18	101	7
Vehicle	19	845	4
Heart	12	918	2
BreastCancer	11	698	2
Ionosphere	35	351	2
Lung Cancer	57	32	3
Dermatology	35	366	6
Sonar	61	208	2
BreastCancerEW	33	569	2
SoybeanSmall	18	13611	7
Movementlibras	91	359	15
Parkinson	24	159	2
Spam	58	4601	2
Waveform	41	4999	3
Arrhythmia	280	452	16
CNAE	857	1079	9
DNA	181	3185	3
Hillvalley	101	1212	2
Vowel	11	900	10

4. Experimental results and discussions

The performance of LO, ELO-U, and ELO-S algorithms are discussed in this section using 21 various datasets with different shapes. In Eq. 4, the fitness function is used with $\alpha = 0.99$ and $\beta = 0.01$, which is original from (Sihwail et al., 2020). The *thres* value is set to 0.5 for the LSA and binary conversion parameter, and the upper and lower bounds' value is pointed to be [0,1], in order to perform the normalization into binary problem (Tubishat et al., 2020; Sihwail et al., 2020). For comparison fairness, Max_{iter} is in 50 iterations, and the N is 10 population size. Such adjustments are previously used by (Hegazy et al., 2019; Tubishat et al., 2020; Sihwail et al., 2020). Final outcomes are gathered from the average of 20 runs, and the Max_{Loc_iter} in LSA is recommended to run for 10 iterations in the proposed ELO model. The accuracy for selected feature subsets is measured based on KNN classifier with a $k = 5$ value.

The evaluation and comparison between LO, ELO-U, and ELO-S have been made using several measurements, such as the average of selected features, and fitness values ($mean, STD$). The *STD* test is used to assess the algorithm's stability in predicting the class and selecting the best feature subset over the iteration in the 20 runs as an average of accuracy and an average of F1_scores. We analyzed the original LO along with two new proposed enhancements ELO-U, and ELO-S algorithm, by examining their respective rates in reaching the optimal solution, and in enhancing diversity by utilizing OBL during the initial population stage.

A comparative process for our proposed ELO model also has been performed against other previous algorithms in literature such as SSA, ISSA, PSO, GA, GWO, and ALO (Hegazy et al., 2020) using the average of *Acc*, the average of selected features, the mean of fitness value *Fit* in 20 runs. Then, the second part of experiments involved ranking the algorithms based on Friedman ranking test (Friedman, 1940; Smucker et al., 2007), and the Wilcoxon ranking one (Woolson, 2007; Rey and Neuhäuser, 2011) in terms of pairwise comparison between the best-proposed method (ELO-U algorithm), and with other competitive previous algorithms.

The experimental evaluation test used four metrics *Acc* in Eq. 12, *Precision* in Eq. 13, *Recall* in Eq. 14, and *F1_score* in Eq. 15:

$$Acc = \frac{TP + TN}{TP + FP + TN + FN} \quad (12)$$

$$Precision = \frac{TP}{TP + FP} \quad (13)$$

$$Recall = \frac{TP}{TP + FN} \quad (14)$$

$$F1\_score = \frac{2 \times Recall \times Precision}{Recall + Precision} \quad (15)$$

where:

1. True Positive (*TP*): The current value is positive, and our proposed model also predicted a positive value.
2. True Negative (*TN*): The current value is negative and our proposed model predicted a negative value.

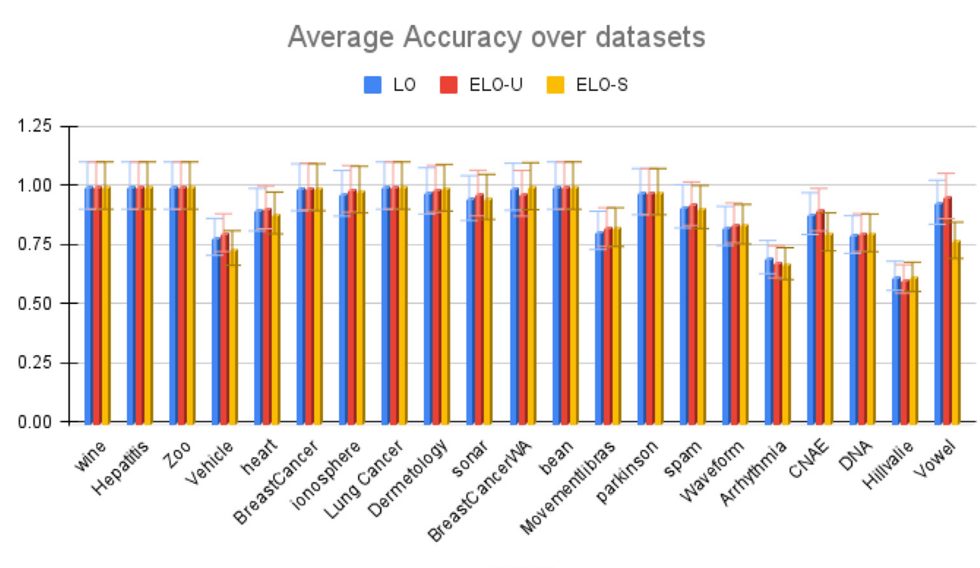

Fig. 8. Average accuracy between LO, ELO-U, ELO-S over Datasets.

Table 3
Accuracy comparisons between LO, ELO-U, ELO-S over datasets.

Dataset	LO	ELO-U	ELO-S
Wine	1	1	1
Hepatitis	1	1	1
Zoo	1	1	1
Vehicle	0.78304	**0.80079**	0.73570
Heart	0.89783	**0.90817**	0.88406
BreastCancer	0.99286	**0.99585**	0.99286
Ionosphere	0.96739	**0.98592**	0.98292
Lung Cancer	1	1	1
Dermatology	0.97748	0.98649	**0.99099**
Sonar	0.94712	**0.96825**	0.95238
BreastCancerEW	0.99476	0.96781	**0.99708**
SoybeanSmall	1	1	1
Movementlibras	0.81019	**0.82407**	**0.82407**
Parkinson	**0.97449**	0.97436	0.97436
Spam	0.91205	**0.92363**	0.90916
Waveform	0.82933	**0.842**	0.83633
Arrhythmia	**0.69773**	0.67803	0.67033
CNAE	0.88117	**0.89815**	0.80429
DNA	0.79278	**0.80115**	0.80005
Hillvalley	**0.61866**	0.60533	0.61454
Vowel	0.92963	**0.95556**	0.76852
Average	0.905070	**0.910267**	0.892268

3. False Positive (*FP*): The current value is negative but our proposed model predicted a positive value.
4. False Negative (*FN*): The model's prediction was negative, while the actual value was positive.

4.1. Experiment 1: Comparative results between LO, ELO-U, ELO-S

We compared LO with our proposed two enhancement algorithms (ELO-U and ELO-S) to demonstrate their ability to accurately detect the class with the least number of selected features. Additionally, we compared LO, ELO-U, and ELO-S using averages of *Fit*, and *STD* for the best fitness value that runs 20 times to determine the stability of the three versions of our proposed models.

Fig. 8 and Table 3 show the average accuracy results of LO, ELO-U, and ELO-S across all datasets. Our findings reveal that ELO-U outperformed other algorithms in 11 datasets. LO achieved the highest accuracy in just three datasets which are Parkinson, Arrhythmia, and Hillvalley datasets. While ELO-S obtained the most accurate results in three datasets, which are Dermatology, BreastCancerEW, and Movementlibras datasets. When comparing the average accuracy across all datasets, ELO-U had the highest value of 0.910267, which is highlighted in bold text to indicate its superiority.

Table 4 displays the average matrices, including precision, recall, and F1_score, which are computed using Eqs. (13)–(15). The results of precision metrics for used datasets across the compared algorithms indicate that our proposed ELO-U algorithm also achieved the best recall and F1_score results, while LO obtained the highest precision value.

The wrapping method leads to reducing the dataset size, and to a minimal subset of features, in order to obtain higher accuracy in detection by evaluating the performance based on the selected feature size across the datasets. Table 5 and Fig. 9 show that our proposed ELO-U algorithm obtained the highest average rank when compared against other algorithm's in overall used datasets. The

Table 5
Average of selected feature among the dataset.

Dataset	LO	ELO-U	ELO-S
Wine	5.5	**4.2**	5.25
Hepatitis	5.75	5	**4.75**
Zoo	5.33	**5**	6
Heart	8.5	**6.333**	8
BreastCancer	4.33	4.333	**3.6667**
Ionosphere	13.5	15	**10.667**
LungCancer	16	18.667	**12.333**
Dermatology	14	**11.333**	11.5
Sonar	24.75	26	**22**
BreastEW	12.75	**12**	12.5
Bean	2.75	3	**1.5**
Movementlibras	38.5	**35.333**	38.25
Parkinson	7.5	**5.5**	6.5
Spam	32.25	**26.75**	27.25
Waveform	21.75	21.333	**21.25**
Arrhythmia	132.5	115	**111.25**
CNAE	423.75	**347.667**	408.75
DNA	87.75	**79**	82.25
Hillvalley	44.25	**41.5**	46
Vehicle	11.25	10.75	**10.5**
Vowel	6.75	7.5	**5.5**
Average	43.7815	**38.15239**	40.74603

Table 4
Comparative results for LO, ELO-U, ELO-S algorithms based on Precision, Recall, and F1_score measures.

	LO			*ELO-U*			*ELO-S*		
Dataset	**Precision**	**Recall**	**F1_score**	**Precision**	**Recall**	**F1_score**	**Precision**	**Recall**	**F1_score**
Wine	1	1	1	1	1	1	1	1	1
Hepatitis	1	1	1	1	1	1	1	1	1
Zoo	1	1	1	1	1	1	1	1	1
Heart	0.89731	0.89601	0.89655	0.90165	0.90200	0.90098	0.88550	0.87988	0.88197
BreastCancer	0.99462	0.98958	0.99203	0.99462	0.98958	0.99203	0.99462	0.98958	0.99203
Ionosphere	0.96864	0.94141	0.95266	0.97656	0.95490	0.96504	0.98936	0.98	0.98442
LungCancer	1	1	1	1	1	1	1	1	1
Dermatology	0.97458	0.975	0.97457	0.98569	0.98611	0.98483	0.98989	0.99074	0.98987
Sonar	0.95067	0.93378	0.94644	0.97092	0.96742	0.96806	0.95833	0.95075	0.95205
BreastEW	0.99588	0.99292	0.99436	0.96020	0.95258	0.95549	0.99771	0.99603	0.99684
Bean	1	1	1	1	1	1	1	1	1
Movementlibras	0.85799	0.81444	0.80387	0.86715	0.82778	0.81475	0.86985	0.82666	0.81447
Parkinson	0.95330	0.98311	0.96678	0.95707	0.98276	0.96803	0.95455	0.98275	0.96741
Spam	0.92003	0.92591	0.92260	0.83603	0.91885	0.91982	0.90411	0.90689	0.90534
Waveform	0.82931	0.82952	0.82925	0.83831	0.83587	0.83577	0.83718	0.83639	0.83650
Arrhythmia	0.56169	0.45177	0.47197	0.52979	0.44025	0.45965	0.61733	0.47331	0.49783
CNAE	0.88907	0.88117	0.88106	0.90355	0.89815	0.89846	0.76655	0.74074	0.74035
DNA	0.79418	0.83036	0.79150	0.80227	0.83378	0.79895	0.79481	0.82993	0.79742
Hilvalley	0.62189	0.61806	0.61647	0.59829	0.59567	0.59270	0.61912	0.61492	0.61134
Vehicle	0.77787	0.78336	0.77846	0.79831	0.80141	0.79641	0.73045	0.73558	0.73141
Vowel	0.93496	0.92963	0.92943	0.95814	0.95370	0.95310	0.78002	0.76851	0.76606
Average	**0.90105**	0.89410	0.89276	0.89898	**0.897134**	**0.895434**	0.889973	0.881082	0.879302

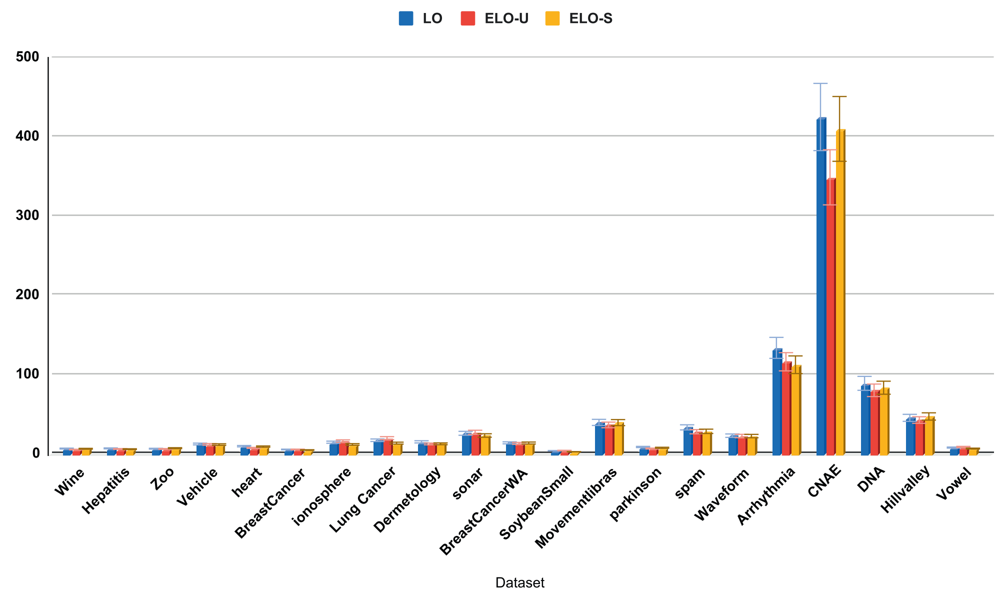

Fig. 9. Average of selected features over all datasets.

Table 6
Comparing fitness value mean, STD, Worst value for LO, ELO-U, ELO-S.

	Mean			*STD*			*Worst*		
Dataset	**LO**	**ELO-U**	**ELO-S**	**LO**	**ELO-U**	**ELO-S**	**LO**	**ELO-U**	**ELO-S**
Wine	0.0047	**0.0042**	0.0044	0.0004	**0.0000**	0.0004	**0.01639**	0.043056	0.040833
Hepatitis	0.0034	**0.00175**	0.0025	0.0003	**0.0002**	**0.0002**	**0.066875**	0.12953	0.127785
Zoo	0.0033	0.00292	**0.0028**	**0.0003**	**0.0003**	**0.0003**	**0.02030**	0.03643	0.051517
Vehicle	0.2207	**0.2044**	0.2133	0.0093	0.0075	**0.0048**	0.263121	0.26544	**0.252332**
Heart	0.1089	**0.1035**	0.10591	0.0054	**0.0037**	**0.0037**	0.14846	**0.12097**	0.13817
BreastCancer	0.01248	**0.0121**	**0.0121**	0.0006	**0.0000**	0.0037	**0.01484**	0.01845	0.022750
Ionosphere	0.0457	0.0172	**0.0171**	0.0055	0.0012	**0.0001**	0.078199	**0.042183**	0.051871
Lung Cancer	0.00286	**0.00113**	0.00220	0.0004	**0.0000**	0.0004	**0.114821**	0.169285	0.115417
Dermetology	0.0262	**0.0131**	0.0132	0.0061	**0.0054**	0.0061	0.058867	**0.05856**	0.06383
Sonar	0.05635	0.03576	**0.05081**	0.0209	0.01147	**0.0002**	**0.152423**	0.15440	0.16225
BreastCancerWA	0.00919	**0.00456**	0.00678	**0.0034**	0.00410	0.00370	**0.0288141**	0.03406	0.03127
SoybeanSmall	0.000882	**0.00058**	0.000784	0.0002	**0.0000**	0.0003	**0.008914**	0.01033	0.00912
Movementlibras	0.19240	**0.17824**	0.178351	0.01341	0.00685	**0.00670**	0.23420	0.22966	**0.22947**
Parkinson	0.02878	**0.01934**	0.02871	0.0086	**0.00021**	0.00042	0.09132	**0.063928**	0.09777
Spam	0.07875	**0.07837**	0.08461	0.0088	**0.00431**	0.00959	0.117895	0.11629	**0.11330**
Waveform	0.17446	0.168107	**0.16728**	0.0028	0.00695	**0.00260**	0.21175	**0.20680**	0.24251
Arrhythmia	0.303219	**0.276498**	0.29640	0.0116	0.0131	**0.00936**	**0.35586**	0.36454	0.36117
CNAE	0.122647	**0.10553**	0.123842	0.0150	0.01711	**0.00391**	0.279745	0.17159	**0.16842**
DNA	0.209712	**0.188803**	0.19831	0.0068	**0.00324**	0.01682	0.290587	0.29116	**0.28350**
Hillvalley	0.381997	**0.36956**	0.38617	0.01672	**0.00683**	0.01116	0.44341	0.44079	**0.43953**
Vowel	0.076333	**0.0535**	**0.0535**	0.00239	**0.002121**	**0.002121**	0.11883	**0.09833**	0.123167

bold text indicates the better result, which leads to the minimum size of subset.

We compared the performance of our proposed ELO-U algorithm to others based on their fitness value, denoted by *Fit*. Table 6 presents the mean of best *Fit* values, the *STD* respect to the best fitness values, and the worst value of *Fit* for each algorithm, which are calculated over 20 runs. The final results indicate that our ELO-U algorithm obtained the best mean fitness value in 15 datasets out of used 21 datasets, followed by ELO-S in 4 datasets, in Vowel, and BreastCancer datasets both ELO-U and ELO-S obtained the same fitness value. Moreover, the *STD* results in the second part of Table 6 report that our proposed ELO-U algorithm is superior with respect to the stability for obtaining the best fitness value across multiple runs.

The convergence rate curves of LO, ELO-U, and ELO-S are depicted in Figs. 10 and 11, showing the average fitness value over each iteration for 20 runs across most datasets in this study. These graphs reveal that our proposed ELO-U algorithm exhibits a convergence trend comparable to other competitive algorithms, which are analyzed in this work.

Another experiment is applied to prove the effectiveness of two proposed improvements from LO algorithm, which are the modified LO-based utilizing Opposition Based Learning called (LO-OBL), and the LO-LSA which stands for Local Search Algorithm. In order to compare their results, we implemented each improvement of LO-OBL and LO-LSA separately, and analyzed the data as shown in Table 7. First, we applied OBL technique only in the LO-OBL version. This approach showed a slight improvement in LO's performance across all criteria. Second, we implemented the LSA strategy on LO without OBL technique. The results showed better improvement in LO's performance across most of used datasets. Lastly, we combined the two improvements in ELO, which ranked as the top performer among LO, LO-OBL, and LO-LSA in all three criteria: $Avg(Fit)$, $Avg(ACC)$, and $Avg(selected\_set)$.

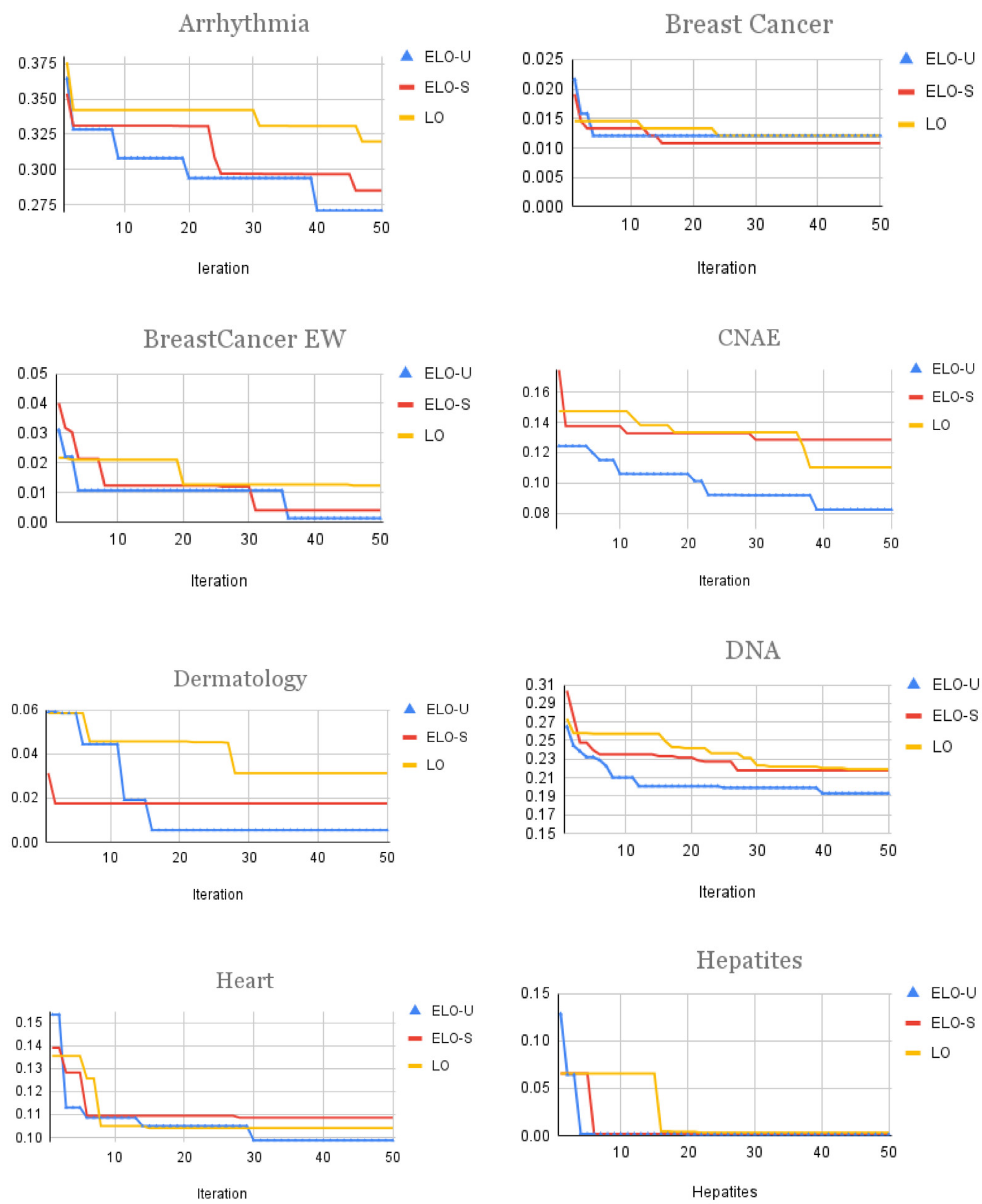

Fig. 10. Convergence example for ELO-U, ELO-S, LO algorithms.

Meanwhile, Table 7 displays the results of using OBL, which are achieved through Eq. 7. This OBL method obtained more effective outputs in selecting solutions than the standard random method. This is because OBL selected the best available solutions while minimizing the selection of weaker solutions. Moreover, the proposed algorithm is capable to balance exploration and exploitation, which are improved using LSA. U-shape TF is further utilized to modify the positions of search agents, which increases algorithm exploration ability. Then, our proposed algorithm will prevent getting stuck in a local solution by locating the promising regions. The proposed neighbourhood search also improves the ability of exploitation to search for Free Risk in the indicated local region. So, our enhanced proposed ELO-U algorithm is superior in three aspects: average classification accuracy $Avg(ACC)$, average fitness value $Avg(Fit)$, and average selected features $Avg(selected\_set)$.

In conclusion, the obtained results from the first experiment of our proposed ELO-U algorithm are the best values in the average of Acc value, in identifying all actual positive rates in the dataset classes, which refers to Recall value (Miao and Zhu, 2022). Our proposed ELO-U algorithm achieved the best results compared with the LO, ELO-S. Also, ELO-U obtained the best values with a minimal size of selected features. In following Experiment 2, we compare and link our proposed ELO-U algorithm with other competitive algorithms used as wrapper methods (SSA, ISSA, PSO, GA, and GWO) that are described in the paper (Hegazy et al., 2020).

4.2. Experiment 2: Comparatives with previous algorithms

In this section, a comparative experiment of our proposed ELO-U algorithm against other meta-heuristic algorithms is conducted, including SSA, ISSA, PSO, GA, and GWO, as listed in (Hegazy et al., 2020). The process of comparisons has been performed in the average of specified features, mean fitness values, and average accuracy.

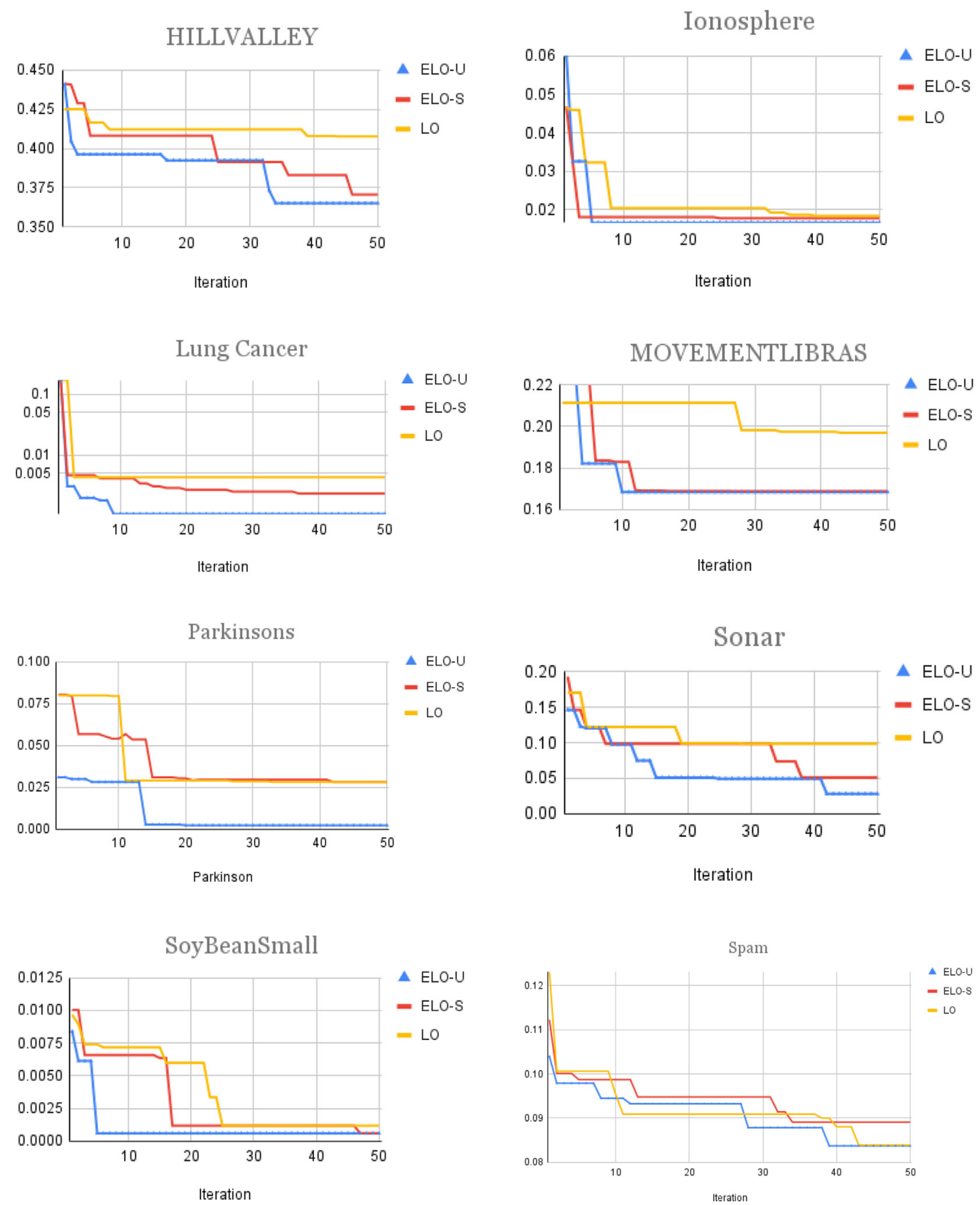

Fig. 11. Convergence example for ELO-U, ELO-S, LO algorithms.

Table 8 displays the average accuracy of comparative wrapper methods overall used datasets. Our proposed ELO-U algorithm outperforms all comparative algorithms in 11 datasets. However, it achieved the same accuracy as the LO algorithm in Wine, Hapitites, Zoo, and Soybean datasets. Fig. 12 shows the average accuracy, which demonstrates the superiority of our proposed ELO-U algorithm. Moreover, the last row in Table 8 illustrates the mean ranking value founded on the Friedman test, conducted using the SPSS tool. The overall *p-value* test resulted in 0.000, indicating that we ignore the null hypothesis, and we suppose that there are variances among comparative algorithms. We noticed that the mean rank for our proposed ELO-U algorithm is the highest, at **7.43**.

The results of variant classification accuracy using our proposed ELO-U algorithm in different datasets such as Wine, Hepatitis, Zoo, and Sonar assess how well the algorithm balances exploration and exploitation. Consequently, our proposed ELO-U algorithm achieved the highest accuracy, followed by LO and ISSA. These findings suggest that ELO-U can effectively balance global and local search, avoiding getting stuck in a local solution and immature convergence, ultimately improving classification accuracy; this is because of imply the OBL and LSA.

The final outcomes comparing the mean of selected features among other competitive algorithms are illustrated in Table 9, which proved the superiority of ISSA with 11 datasets and an average of 29.255 overall datasets. The Friedman mean rank obtained 1.93. ELO-U obtained an average of 38.152 overall datasets. The mean rank of Friedman test shows a value of 2.68. Meanwhile, Friedman examined the experiment and it obtained a 0.000 in *p-value*, which shows differences between comparative algorithms, and rejects the null hypothesis. Final outcomes indicate that using

Table 7
Comparing LO, LO-OBL, LO-LSA, ELO-U algorithms.

	Avg(Fit)				*Avg(ACC)*				*Avg(Selected_set)*			
Dataset	**LO**	**LO-OBL**	**LO-LSA**	**ELO-U**	**LO**	**LO-OBL**	**LO-LSA**	**ELO-U**	**LO**	**LO-OBL**	**LO-LSA**	**ELO-U**
Wine	0.00470	0.00542	0.00563	**0.004167**	**1**	0.965278	**1**	**1**	5.5	5	6.75	**4.2**
Hepatitis	0.00342	0.034359	0.00276	**0.001754**	**1**	0.968750	**1**	**1**	5.75	5.5	5.75	**5**
Zoo	0.00333	0.003437	0.00313	**0.002917**	**1**	**1**	**1**	**1**	5.33	6	**5**	**5**
Vehicle	0.22072	0.218943	0.21197	**0.204441**	0.7830375	0.774500	0.665680	**0.800789**	11.25	9.5	7.75	10.75
Heart	0.10888	0.107720	0.10795	**0.103515**	0.8978261	0.810876	0.788040	**0.908174**	8.5	7	**4.75**	6.33
BreastCancer	0.01249	0.013009	**0.01207**	**0.01207**	0.9928571	0.987500	0.99286	**0.995850**	4.33	4	**4**	4.33
Ionosphere	0.04579	0.017620	0.01769	**0.01718**	0.9673864	0.985915	**0.98592**	0.985916	13.5	14	**13**	15
Lung Cancer	0.00286	0.003438	0.00254	**0.00113**	**1**	**1**	**1**	**1**	16	21	**14.25**	18.667
Dermetology	0.02624	0.0311507	0.01701	**0.01306**	0.9774775	0.972973	**0.98649**	**0.98649**	14	16.5	12	**11.33**
Sonar	0.05636	0.0867917	0.06849	**0.03576**	0.9471154	0.916667	0.93452	**0.968254**	**24.75**	28.5	27	26
BreastCancerWA	0.00919	0.016943	0.00567	**0.00456**	0.9947552	0.986842	**0.99780**	0.967811	12.75	12.5	**10.5**	12
SoybeanSmall	0.00088	0.002890	0.00143	**0.00059**	**1**	0.994329	**1**	**1**	2.75	3.5	**1.25**	3
Movementlibras	0.19240	0.183500	**0.17216**	0.17824	0.8101852	0.819444	**0.82986**	0.824074	38.5	43	**33.5**	35.33
Parkinson	0.02878	0.029021	0.02834	**0.01935**	**0.9744898**	0.974359	0.97436	0.974359	7.5	9.5	6.5	**5.5**
Spam	0.07876	0.093828	0.08118	**0.07837**	0.9120521	0.874593	0.90445	**0.923634**	32.25	34	28.5	**26.75**
Waveform	0.17446	0.190028	0.17226	**0.16811**	0.8293333	0.770250	0.80475	**0.842000**	21.75	**20**	21.5	21.33
Arrhythmia	0.30322	0.339318	0.31980	**0.27650**	**0.6977273**	0.662088	0.68132	0.678030	132.5	129.5	120.5	**115**
CNAE	0.12265	0.124945	0.11007	**0.10554**	0.8811728	0.762731	0.83565	**0.898148**	423.75	424.5	427	**347.67**
DNA	0.20971	0.2198093	0.20408	**0.18880**	0.7927786	0.78336	0.79867	**0.801151**	87.75	95.5	85.25	**79**
Hillvalley	0.38200	0.3862444	0.38150	**0.36957**	0.6186557	0.61420	**0.61934**	0.605336	44.25	41.5	46.5	**41.5**
Vowel	0.07633	0.0561250	**0.05313**	0.0535000	0.9296296	0.802778	0.93056	**0.955560**	6.75	**4**	7.5	7.5
Average					0.9050705	0.87750	0.891920	**0.910265**	43.781	44.5	42.321	**38.152**

Table 8
Average classification accuracy in 20 runs.

Dataset	**LO**	**ELO-U**	**ISSA**	**SSA**	**GA**	**PSO**	**ALO**	**GWO**
Wine	**1**	**1**	0.978	0.955	0.957	0.923	0.942	0.912
Hepatitis	**1**	**1**	0.9123	0.8939	0.875	0.8605	0.8825	0.8479
Zoo	**1**	**1**	0.872	0.798	0.854	0.824	0.805	0.875
Vehicle	0.7830	**0.8007**	0.7398	0.7352	0.7163	0.6695	0.6814	0.6172
Heart	0.8978	**0.9081**	0.813	0.813	0.824	0.822	0.802	0.807
BreastCancer	0.9928	**0.9958**	0.957	0.955	0.955	0.951	0.95	0.953
Ionosphere	0.96738	**0.9859**	0.853	0.836	0.824	0.848	0.843	0.819
Lung Cancer	**1**	**1**	0.5978	0.6023	0.482	0.5627	0.5056	0.5014
Dermetology	0.97747	**0.9864**	0.9825	0.9623	0.9645	0.9071	0.9322	0.9488
Sonar	0.94712	**0.9683**	0.734	0.737	0.717	0.723	0.714	0.714
BreastCancerWA	**0.99476**	0.9678	0.961	0.942	0.935	0.949	0.942	0.949
SoybeanSmall	**1**	**1**	0.9708	0.9736	0.9438	0.8648	0.9098	0.9205
Movementlibras	0.81019	**0.82407**	0.6938	0.6899	0.6902	0.6466	0.6597	0.6866
Parkinson	**0.974490**	0.97436	0.8529	0.8429	0.8492	0.8653	0.8367	0.8367
Spam	0.912052	**0.92363**	0.8814	0.8735	0.8229	0.8735	0.8804	0.8839
Waveform	0.82933	**0.842**	0.766	0.769	0.762	0.762	0.769	0.765
Arrhythmia	**0.69773**	0.67803	0.66	0.6378	0.5802	0.5707	0.5462	0.5641
CNAE	0.881173	**0.89815**	0.8912	0.8514	0.8246	0.8147	0.7962	0.8407
DNA	0.79278	0.80115	**0.8611**	0.8429	0.7954	0.7979	0.7666	0.8425
Hillvalley	0.618656	0.60533	**0.6319**	0.6319	0.5627	0.5507	0.5709	0.5544
Vowel	0.929629	**0.95556**	–	–	0.64	0.7	–	–
Average	0.9050704	**0.9102654**	0.830475	0.81713	0.78927	0.785047	0.78676	0.791935
Friedman mean test	6.88	**7.43**	5.75	4.50	3.35	2.83	2.45	2.83

OBL and LSA helps to decrease the number of selected features, and increase the accuracy of classification. Additionally, ELO-U utilizes the U-shape TF to target informative areas in the search space, eliminating irrelevant ones, and selecting necessary features.

Table 11 presents the mean best fitness value among the comparative algorithms for 20 runs over the datasets. The results proved that our proposed ELO-U model obtained the best with 16 datasets, and the average for all datasets is also the best with a value of **0.09281**; the lower fitness value means better performance. The ELO-U algorithm has been shown to have better fitness value than other algorithms in multiple datasets, indicating its solid abilities and lower classification error. The dynamic LSA search also efficiently identifies the best solution and promising areas. Friedman's mean rank test for the comparative algorithms shows that the proposed ELO-U algorithm obtained the best rank with a value of **2.35**, and obtained 0.000 value in *p-value* for all the experiments, indicating that there severe differences among other comparative algorithms.

Comparative process using Wilcoxon signed ranking test, Table 10 illustrates the *p-value* for pairwise comparisons of the algorithms. If the *p-value* $<$ than 0.05, this indicates a serious difference between the compared pair, and the null hypothesis is ignored. Otherwise, if *p-value* $\geqslant$0.05 the null hypothesis is retained, which concludes not much divergence among all compared algorithms.

5. Conclusion

This paper offers new empowered wrapper algorithms for feature selection by developing Lemurs Optimization (LO) Algorithm, which is a new enhancement for LO in two binary new versions:

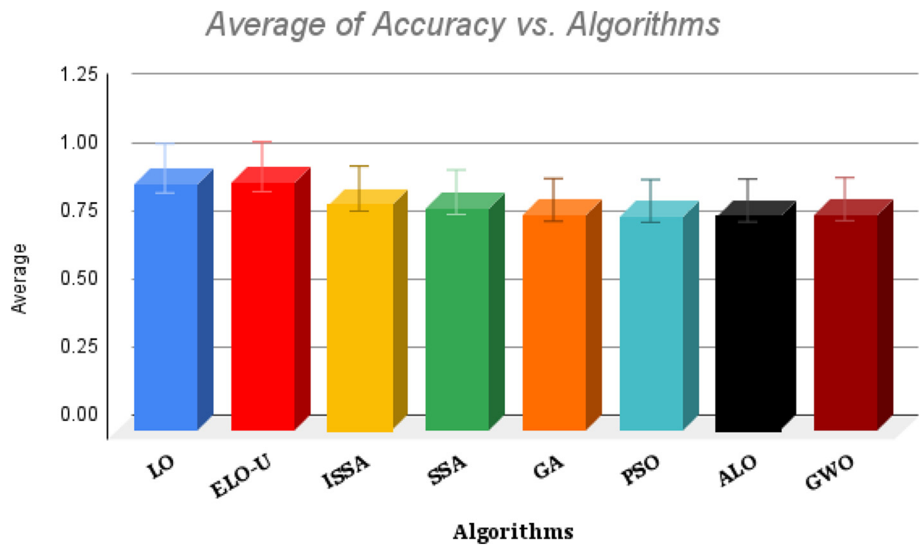

Fig. 12. Average of Accuracy results of proposed ELO-U against other Algorithms.

Table 9
Average selection size for all optimizers in 20 runs.

Dataset	LO	ELO-U	ISSA	SSA	GA	PSO	ALO	GWO
Wine	5.5	**4.2**	4.5	5.07	4.8	6.6	5	7
Hepatitis	5.75	5	**3.4**	4.5	5.1	8.3	5.7	8.2
Zoo	5.33	**5**	5.4	7.9	11.25	6.95	6.3	5.2
Vehicle	11.25	10.75	**8.4**	**8.4**	12.3	11.05	**8.4**	11.9
Heart	8.5	6.33	6.2	6.2	6.35	8.4	**5.4**	6.3
BreastCancer	4.33	4.33	**4.02**	5.9	5.3	5.4	6.1	5.37
Ionosphere	13.5	15	**11.07**	16.08	16.47	16.78	13.6	21.04
Lung Cancer	**16**	18.667	20.45	22.06	20.42	27.8	25.47	28.04
Dermetology	14	**11.33**	17.39	21.27	20.48	25.7	23.47	24.03
Sonar	24.75	26	18.89	**17.49**	29.78	31.7	32.11	29.94
BreastCancerEW	12.75	**12**	12.91	18.04	15.25	15.47	13.74	15.05
SoybeanSmall	**2.75**	3	9.04	13.26	15.23	14.79	21.49	20.79
Movementlibras	38.5	35.33	**23.78**	35.48	35.02	41.04	39.48	42.49
Parkinson	7.5	**5.5**	5.51	6.79	7.25	9.48	11.23	5.28
Spam	32.25	26.75	**23.78**	29.48	31.47	29.06	29.45	32.45
Waveform	21.75	21.33	**19.78**	23.71	29.8	30	29.45	32.54
Arrhythmia	132.5	115	**95.15**	109.48	125.4	119.95	113.27	186
CNAE	423.75	347.67	**216.23**	225.2	327.14	573.14	514.45	522.05
DNA	87.75	79	**59.8**	65.2	118.45	112.6667	139.8	89.8
Hillvalley	44.25	41.5	**19.4**	26.2	44	52.57	45.64	57.4
Vowel	6.75	7.5	–	–	–	–	–	–
Average	45.6330	38.1520	**29.2550**	33.3860	44.0630	57.3420	54.4780	57.5440
Friedman mean test	4.22	2.68	**1.93**	3.93	5.10	6.50	5.30	6.35

Table 10
Wilcoxon pairwise result among ELO-U and comparative optimizers.

	Average features	**of selected**	**Mean of best** *Fit*		**Average of** *Acc*	
Pairs	*p-value*	**significant differences**	*p-value*	**significant differences**	*p-value*	**significant differences**
ELO-U vs. LO	0.012	Yes	0.033	Yes	0.121	No
ELO-U vs. ISSA	0.044	Yes	0.167	No	0.001	Yes
ELO-U vs. SSA	0.765	No	0.126	No	0.000	Yes
ELO-U vs. GA	0.002	Yes	0.033	Yes	0.000	Yes
ELO-U vs. PSO	0.000	Yes	0.006	Yes	0.000	Yes
ELO-U vs. ALO	0.002	Yes	0.006	Yes	0.000	Yes
ELO-U vs. GWO	0.000	Yes	0.009	Yes	0.000	Yes

ELO-U and ELO-S. Basically, LO algorithm is based on a set of cooperative lemurs inspired by their social behaviors, while our new enhanced versions is developed by adapting Opposition Based Learning (OBL) and Local Search Algorithm (LSA). They combined with two transfer functions (U-shape TF as 'ELO-U' and Sigmoid TF as 'ELO-S') functions. The performances of the three algorithms (LO, ELO-U, and ELO-S) are compared using 21 UCI datasets by utilizing various evaluation metrics, like accuracy, fitness value,

Table 11
Mean statistical fitness values in 20 runs for all optimizer algorithms.

Dataset	ISSA	SSA	GA	PSO	ALO	GWO	LO	ELO-U
Wine	0.01	0.007	0.023	0.054	0.0372	0.043	0.00470	**0.00444**
Hepatitis	0.11	0.12	0.16	0.15	0.1293	0.2195	0.00342	**0.00246**
Vehicle	0.23	0.21	0.23	0.21	0.3019	0.2945	0.00333	**0.00275**
Zoo	**0.059**	0.079	0.116	0.2	0.1875	0.1027	0.22072	0.21336
Heart	0.12	0.116	0.125	0.185	0.2026	0.1301	0.10888	**0.10591**
Breastcancer	0.019	0.039	0.038	0.037	0.0425	0.0415	0.01249	**0.01207**
Ionosphere	0.097	0.118	0.127	0.176	0.1199	0.1415	0.04579	**0.01708**
Lung cancer	0.09	0.07	0.15	0.1724	0.1902	0.2844	0.00286	**0.00220**
Dermatlogy	0.003	0.03	0.09	0.0425	0.0214	0.0226	0.02624	**0.01316**
Sonar	0.145	0.137	0.169	0.319	0.1769	0.2256	0.05636	**0.05081**
BreastCancerEW	0.024	0.032	0.042	0.054	0.0564	0.0654	0.00919	**0.00678**
SoybeanSmall	0.03	0.04	0.06	0.04	0.0019	0.2913	0.00088	**0.00078**
Movementlibras	0.227	0.29	0.296	0.333	0.3889	0.2784	0.19240	**0.17835**
Parkinsons	0.032	0.055	0.084	0.095	0.0936	0.1242	0.02878	**0.02872**
Spambase	0.239	0.218	0.221	0.276	0.2904	0.284	**0.07876**	0.08461
Waveform	0.209	0.208	0.218	0.24	0.2681	0.2648	0.17446	**0.16728**
Arrhythmia	0.044	0.046	0.06	0.09	0.0374	**0.0363**	0.30322	0.29641
CNAE	0.209	0.251	0.267	0.327	0.1662	0.2	0.12265	**0.12384**
DNA	**0.119**	0.155	0.178	0.191	0.1371	0.1452	0.20971	0.19831
Hillvalley	**0.215**	0.225	0.245	0.263	0.4052	0.4088	0.38200	0.38617
Vowel	–	–	–	–	–	–	0.07633	**0.05350**
Average	0.11155	0.13824	0.14495	0.172745	0.16273	0.18019	0.09825	**0.09281**
Friedman mean rank	3.28	3.90	5.43	6.25	5.80	6.15	2.85	**2.35**

F1_scores, and feature selection size. The final outcomes showed that our proposed ELO-U algorithm outperforms the main two algorithms (LO and ELO-S), and outperforms several other wrapper methods that have been published in literature, such as ISSA, SSA, PSO, and GA algorithms. With respect to the fitness value and accuracy metrics, our proposed ELO-U algorithm achieved the highest accuracy overall used datasets compare among other comparative algorithms The obtained results are 80.07% in minimal, and in maximum 100% for 5 datasets, and meanwhile the average accuracy achieved is 91.03%. Additionally, the convergence rate of our enhanced proposed ELO-U algorithm is faster than other algorithms, and it obtained the best results with respect to the mean and fitness STD. Therefore, our proposed ELO-U algorithm can be a promising solution for FS problems in the ML domain.

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.

References

Abasi, A.K., Makhadmeh, S.N., Al-Betar, M.A., Alomari, O.A., Awadallah, M.A., Alyasseri, Z.A.A., Doush, I.A., Elnagar, A., Alkhammash, E.H., Hadjouni, M., 2022. Lemurs optimizer: A new metaheuristic algorithm for global optimization. Appl. Sci. 12, 10057.

Abu Doush, I., Al-Betar, M.A., Awadallah, M.A., Hammouri, A.I., Al-Khatib, R.M., ElMustafa, S., ALkhraisat, H., 2018. Harmony search algorithm for patient admission scheduling problem. J. Intell. Syst. 29, 540–553.

Agrawal, P., Abutarboush, H., Talari, G., Wagdy, A., 2021. Metaheuristic algorithms on feature selection: A survey of one decade of research (2009–2019). IEEE Access PP, 1–1. https://doi.org/10.1109/ACCESS.2021.3056407.

Al-Khatib, R.M., Al-Betar, M.A., Awadallah, M.A., Nahar, K.M., Shquier, M.M.A., Manasrah, A.M., Doumi, A.B., 2019. Mga-tsp: modernised genetic algorithm for the travelling salesman problem. Int. J. Reason.-based Intell. Syst. 11, 215–226.

Al-Khatib, R.M., El-Omari, N.K.T., Al-Betar, M.A., 2023. Innovative cloud computing object-oriented model to unify heterogeneous data. Int. J. Oper. Res. 46, 289–322.

Alaiad, A., Migdady, A., Al-Khatib, R.M., Alzoubi, O., Zitar, R.A., Abualigah, L., 2023. Autokeras approach: A robust automated deep learning network for diagnosis disease cases in medical images. J. Imag. 9, 64.

Alkhateeb, F., Al-Khatib, R.M., Doush, I.A., 2020. A survey for recent applications and variants of nature-inspired immune search algorithm. Int. J. Comput. Appl. Technol. 63, 354–370.

Barhoush, M., Abed-alguni, B.H., Al-qudah, N.E.A., 2023. Improved discrete salp swarm algorithm using exploration and exploitation techniques for feature selection in intrusion detection systems. J. Supercomput., 1–45

Cai, J., Luo, J., Wang, S., Yang, S., 2018. Feature selection in machine learning: A new perspective. Neurocomputing 300, 70–79.

Chawla, N.V., Bowyer, K.W., Hall, L.O., Kegelmeyer, W.P., 2002. Smote: synthetic minority over-sampling technique. J. Artif. Intell. Res. 16, 321–357.

Chen, Y., Liu, J., Zhu, J., Wang, Z., 2023. An improved binary particle swarm optimization combing v-shaped and u-shaped transfer function. Evolutionary. Intelligence.

Ervural, B., Hakli, H., 2023. A binary reptile search algorithm based on transfer functions with a new stochastic repair method for 0–1 knapsack problems. Comput. Ind. Eng. 109080.

Friedman, M., 1940. A comparison of alternative tests of significance for the problem of m rankings. Ann. Math. Stat. 11, 86–92.

Gao, Y., Zhou, Y., Luo, Q., 2020. An efficient binary equilibrium optimizer algorithm for feature selection. IEEE Access 8, 140936–140963.

Guha, R., Ghosh, K.K., Bera, S.K., Sarkar, R., Mirjalili, S., 2023. Discrete equilibrium optimizer combined with simulated annealing for feature selection. J. Comput. Sci. 101942.

Gupta, N., Jain, R., Gupta, D., Khanna, A., Khamparia, A., 2020. Modified ant lion optimization algorithm for improved diagnosis of thyroid disease. In: Cognitive Informatics and Soft Computing: Proceeding of CISC 2019. Springer, pp. 599–610.

Gupta, S., Deep, K., 2019. A hybrid self-adaptive sine cosine algorithm with opposition based learning. Expert Syst. Appl. 119, 210–230.

Hegazy, A.E., Makhlouf, M., El-Tawel, G.S., 2019. Feature selection using chaotic salp swarm algorithm for data classification. Arabian J. Sci. Eng. 44, 3801–3816.

Hegazy, A.E., Makhlouf, M., El-Tawel, G.S., 2020. Improved salp swarm algorithm for feature selection. J. King Saud Univ.-Comput. Infr. Sci. 32, 335–344.

Hernandez-Castro, J.C., Tapiador, J.E., Peris-Lopez, P., Clark, J.A., Talbi, E.G., 2009. Metaheuristic traceability attack against slmap, an rfid lightweight authentication protocol. In: 2009 IEEE International Symposium on Parallel & Distributed Processing. IEEE, pp. 1–5.

Hu, P., Pan, J.S., Chu, S.C., 2020. Improved binary grey wolf optimizer and its application for feature selection. Knowl.-Based Syst. 195, 105746.

Ibrahim, H.T., Mazher, W.J., Uçan, O.N., Bayat, O., 2017. Feature selection using salp swarm algorithm for real biomedical datasets.

Jolly, A., 1966. Lemur social behavior and primate intelligence: The step from prosimian to monkey intelligence probably took place in a social context. Science 153, 501–506.

Mafarja, M., Jarrar, R., Ahmad, S., Abusnaina, A.A., 2018. Feature selection using binary particle swarm optimization with time varying inertia weight strategies. In: Proceedings of the 2nd International Conference on Future Networks and Distributed Systems, pp. 1–9.

Mafarja, M.M., Mirjalili, S., 2017. Hybrid whale optimization algorithm with simulated annealing for feature selection. Neurocomputing 260, 302–312.

Miao, J., Zhu, W., 2022. Precision–recall curve (prc) classification trees. Evol. Intell. 15, 1545–1569.

Michalewicz, Z., 1994. Genetic Algorithms + Data Structures = Evolution Programs. Springer-Verlag, Berlin, Heidelberg.

Mirjalili, S., Gandomi, A.H., Mirjalili, S.Z., Saremi, S., Faris, H., Mirjalili, S.M., 2017. Salp swarm algorithm: A bio-inspired optimizer for engineering design problems. Adv. Eng. Softw. 114, 163–191.

Mirjalili, S., Mirjalili, S.M., Lewis, A., 2014. Grey wolf optimizer. Adv. Eng. Softw. 69, 46–61.
Mirjalili, S., Zhang, H., Mirjalili, S., Chalup, S., Noman, N., 2020. A novel u-shaped transfer function for binary particle swarm optimisation. Soft Computing for Problem Solving 2019: Proceedings of SocProS 2019, vol. 1. Springer, pp. 241–259.
Moradi, P., Gholampour, M., 2016. A hybrid particle swarm optimization for feature subset selection by integrating a novel local search strategy. Appl. Soft Comput. 43, 117–130.
Nguyen, B.H., Xue, B., Zhang, M., 2020. A survey on swarm intelligence approaches to feature selection in data mining. Swarm Evol. Comput. 54, 100663.
Olorunda, O., Engelbrecht, A.P., 2008. Measuring exploration/exploitation in particle swarms using swarm diversity. In: 2008 IEEE Congress on Evolutionary Computation (IEEE World Congress on Computational Intelligence). IEEE, pp. 1128–1134.
Pan, J.S., Hu, P., Snasel, V., Chu, S.C., 2022. A survey on binary metaheuristic algorithms and their engineering applications. Artif. Intell. Rev.
Powzyk, J.A., Mowry, C.B., 2007. The feeding ecology and related adaptations of indri indri. Lemurs: Ecol. Adapt., 353–368
Rahnamayan, S., Tizhoosh, H.R., Salama, M.M., 2007. A novel population initialization method for accelerating evolutionary algorithms. Comput. Mathe. Appl. 53, 1605–1614.
Rey, D., Neuhäuser, M., 2011. Wilcoxon-signed-rank test. In: International Encyclopedia of Statistical Science. Springer, pp. 1658–1659.
Sihwail, R., Omar, K., Ariffin, K.A.Z., Tubishat, M., 2020. Improved harris hawks optimization using elite opposition-based learning and novel search mechanism for feature selection. IEEE Access 8, 121127–121145.
Smucker, M.D., Allan, J., Carterette, B., 2007. A comparison of statistical significance tests for information retrieval evaluation. In: Proceedings of the Sixteenth ACM Conference on Conference on Information and Knowledge Management, pp. 623–632.
Song, X., Zhang, Y., Gong, D., Liu, H., Zhang, W., 2022. Surrogate sample-assisted particle swarm optimization for feature selection on high-dimensional data. IEEE Trans. Evol. Comput.
Thawkar, S., Sharma, S., Khanna, M., kumar Singh, L., 2021. Breast cancer prediction using a hybrid method based on butterfly optimization algorithm and ant lion optimizer. Comput. Biol. Med. 139, 104968.
Tizhoosh, H.R., 2005. Opposition-based learning: a new scheme for machine intelligence. In: International conference on computational intelligence for modelling, control and automation and international conference on intelligent agents, web technologies and internet commerce (CIMCA-IAWTIC'06). IEEE, pp. 695–701.
Tubishat, M., Idris, N., Shuib, L., Abushariah, M.A., Mirjalili, S., 2020. Improved salp swarm algorithm based on opposition based learning and novel local search algorithm for feature selection. Expert Syst. Appl. 145, 113–122.
Tubishat, M., Ja'afar, S., Alswaitti, M., Mirjalili, S., Idris, N., Ismail, M.A., Omar, M.S., 2021. Dynamic salp swarm algorithm for feature selection. Expert Syst. Appl. 164, 113–873.
Woolson, R.F., 2007. Wilcoxon signed-rank test. Wiley Encyclopedia Clin. Trials, 1–3.
Xue, B., Zhang, M., Browne, W.N., Yao, X., 2015. A survey on evolutionary computation approaches to feature selection. IEEE Trans. Evol. Comput. 20, 606–626.
Zhang, R., Nie, F., Li, X., Wei, X., 2019. Feature selection with multi-view data: A survey. Infr. Fusion 50, 158–167.