

A Self-Adapting and Efficient Dandelion Algorithm and Its Application to Feature Selection for Credit Card Fraud Detection

Honghao Zhu , Member, IEEE, MengChu Zhou , Fellow, IEEE, Yu Xie , and Aiiad Albeshri 

Abstract—A dandelion algorithm (DA) is a recently developed intelligent optimization algorithm for function optimization problems. Many of its parameters need to be set by experience in DA, which might not be appropriate for all optimization problems. A self-adapting and efficient dandelion algorithm is proposed in this work to lower the number of DA's parameters and simplify DA's structure. Only the normal sowing operator is retained; while the other operators are discarded. An adaptive seeding radius strategy is designed for the core dandelion. The results show that the proposed algorithm achieves better performance on the standard test functions with less time consumption than its competitive peers. In addition, the proposed algorithm is applied to feature selection for credit card fraud detection (CCFD), and the results indicate that it can obtain higher classification and detection performance than the state-of-the-art methods.

Index Terms—Credit card fraud detection (CCFD), dandelion algorithm (DA), feature selection, normal sowing operator.

I. INTRODUCTION

OPTIMIZATION problems exist in daily life, and can be solved by using mathematical algorithms. When their scales grow, though, they become incredibly challenging to solve. It is found that meta-heuristic algorithms are a class of efficient algorithms to solve such complex problems. In the past decades, numerous meta-heuristics algorithms were proposed, such as genetic algorithms (GA) [1], artificial bee

colony algorithm (ABC) [2], particle swarm optimizer (PSO) [3], artificial fish swarm algorithm (AFSA) [4], firefly optimization algorithm (FA) [5], cuckoo optimization algorithm (COA) [6], bat algorithm (BA) [7], grey wolf optimization algorithm (GWO) [8], evolution strategy with covariance matrix adaptation (CMA-ES) [9], and fireworks algorithm (FWA) [10]. Although they have achieved some success in some fields, the free lunch theory [11] indicates that an algorithm cannot perform optimally on all problems, and thus many try to propose new algorithms to face new challenges.

A dandelion algorithm (DA) is a recently proposed swarm intelligence algorithm as inspired by dandelion sowing [12]. Due to its excellent performance in function optimization, DA has attracted the attention of many scholars, and they have proposed many variants of it to further improve its performance. Gong *et al.* [13] propose an improved dandelion algorithm called MDA, in which Levy mutation is replaced by the mean position of all seeds. They also give the proof of convergence and analysis of parameters. Zhu *et al.* [12] propose a dandelion algorithm with probability-based mutation, in which three probability models, namely linear (DAPM_L), binomial (DAPM_B), and exponential (DAPM_E) models, are designed to select whether to utilize Levy or Gaussian mutation. Han and Zhu [14] propose a fusion dandelion algorithm-based adaptive selection strategy (FSS-DA) in which a distance-aware selection strategy (DSS) by jointly considering the function fitness value and the individual position distance, and other famous selection strategy are put into its fusion pool together. The best selection strategy decided by the overall reward is chosen to select the suitable dandelions for the next generation. To determine whether a seed is excellent or not, an improvement of the evolution process of DA with an extreme learning machine (ELMDA) is proposed [15], where the training model is first built based on the extreme learning machine (ELM) [16] with the training set, which is constructed by the excellent dandelions and poor dandelions assigned their corresponding labels (i.e., +1 if excellent or -1 if poor), and then the training model is applied to classify the seeds into excellent or poor ones. Excellent seeds are chosen to participate in the evolution process, while the poor ones are abandoned to save function evaluations. Han *et al.* [17] propose competition-driven dandelion algorithms in which a novel competition mechanism is designed to find out the loser, and then three historical information models are designed by exploit-

Manuscript received June 23, 2023; revised August 17, 2023 and September 27, 2023; accepted October 2, 2023. This work was supported by the Institutional Fund Projects (IFPIP-1481-611-1443), the Key Projects of Natural Science Research in Anhui Higher Education Institutions (2022 AH051909), the Provincial Quality Project of Colleges and Universities in Anhui Province (2022sdx020, 2022xqhz044), and Bengbu University 2021 High-Level Scientific Research and Cultivation Project (2021pyxm04). Recommended by Associate Editor Xin Luo. (Corresponding authors: MengChu Zhou and Yu Xie.)

Citation: H. Zhu, M. C. Zhou, Y. Xie, and A. Albeshri, "A self-adapting and efficient dandelion algorithm and its application to feature selection for credit card fraud detection," *IEEE/CAA J. Autom. Sinica*, vol. 11, no. 2, pp. 377–390, Feb. 2024.

H. Zhu is with the School of Computer and Information Engineering, Bengbu University, Bengbu 233030, China (e-mail: zhh@bbe.edu.cn).

M. C. Zhou is with the School of Information and Electronic Engineering, Zhejiang Gongshang University, Hangzhou 310018, China, and also with the Helen and John C. Hartmann, the Department of Electrical and Computer Engineering, New Jersey Institute of Technology, Newark NJ 07102 USA (e-mail: zhou@njit.edu).

Y. Xie is with the College of Information Engineering, Shanghai Maritime University, Shanghai 201306, China (e-mail: xieyu@shmtu.edu.cn).

A. Albeshri is with the Department of Computer Science, King Abdulaziz University, Jeddah 21481, Saudi Arabia (e-mail: aalalbeshri@kau.edu.sa).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/JAS.2023.124008

ing historical information with an estimation-of-distribution algorithm [18] to generate a new individual to replace the loser. In addition, they have been applied in many fields, such as traffic flow prediction [19], credit card fraud detection (CCFD) [20], and multi-skill resource-constrained project scheduling [21]. Even if they perform better than their peers, it is found that new parameters are introduced by them. For all these methods, the structures are complex and have a large number of parameters to tune.

Filter, wrap, and embedded methods are commonly used in feature selection. The filter method is completely independent of any machine learning algorithm. It sorts data features according to certain criteria and selects the best feature subset. The wrapped method relies on a specific classification model to evaluate the selected feature subset. Because it directly optimizes the target, the wrapped method performs better. The embedded method imparts feature selection into the classification model and combines the advantages of filtering and wrapping methods [22]. In this paper, a random search package method is used for feature selection, which mainly uses an intelligent optimization algorithm to iteratively add and delete features to establish feature subsets.

Sun *et al.* [23] proposed to use a genetic algorithm to select good feature subsets and support vector machine (SVM) as the base classifier to improve the detection rate. Xue *et al.* [24] proposed three new PSO initialization strategies and three new individual optimal and global optimal update mechanisms for feature selection to achieve the purpose of maximizing classification performance, minimizing the number of features, and reducing computation time. The base classifier used in this method is the K-nearest neighbor. Masood *et al.* [25] proposed a new rank-based incremental search strategy method, Wrack-ELM, which adopted a feature selection method based on package sorting and used the ELM classifier to evaluate the selected features. Xue *et al.* [26] proposed an adaptive particle swarm optimization algorithm for feature selection, especially for large-scale feature selection. KNN is used as the base classifier, and the experimental results show that the design method is effective. Sagban *et al.* [27] combined the bat algorithm and the ant colony algorithm to conduct a feature selection algorithm for the classification of cervical cancer data sets, but this method only takes accuracy as an evaluation index in terms of experimental results. Currently, existing intelligent optimization algorithms primarily focus on feature selection in the context of balanced data classification. Therefore, investigating how to design an improved intelligent algorithm for feature selection that enhances the classification performance on imbalanced data is a worthwhile research problem.

According to the analysis of the literature, the existing literature uses intelligent optimization algorithms to select features for balanced data, and there is a lack of intelligent optimization algorithms for feature selection under unbalanced classification. Among the existing intelligent optimization algorithms, the Dandelion algorithm, as a newly proposed algorithm, has superior performance. It needs to set many

parameters set. Once they are fixed, it may perform well for some problems but no others. Therefore, it is worthwhile to reduce the number of parameters and simplify the Dandelion structure such that it can select more discriminative features and minimize feature redundancy and feature noise to improve the classification results of imbalanced data. In addition, many researchers have used intelligent algorithms to optimize the extreme learning machine to achieve excellent results, but they only deal with balanced data and do not consider the task of imbalanced data. The motivation of this paper stems from the aforementioned problems. Finding solutions address the above-mentioned issues holds significant importance for researchers and financial professionals in recognizing the role of feature selection in detecting credit card fraud given imbalanced classification scenarios.

In order to address these issues well, this work proposes a self-adapting and efficient dandelion algorithm, namely SEDA. Specifically, basic DA consists of normal sowing, mutation sowing, and selection strategy, but only normal sowing is retained to generate seeds in SEDA, i.e., mutation sowing is removed. Meanwhile, the population of SEDA has only one dandelion, i.e., core dandelion, which is different from the basic DA including core dandelion and assistant ones. In addition, an adaptive seeding radius strategy is designed to replace the two parameters (withering factor and growth factor). Finally, the greedy selection strategy is applied to choose the best one in the population into the next generation. Based on the above design, it is noted that all unnecessary mechanisms of DA are removed in SEDA. The proposed algorithm has the following advantages:

- 1) It is easy to implement.
- 2) It has only one parameter, i.e., the number of seeds.
- 3) It reduces time consumption.

The proposed algorithm is used to select proper features for credit card fraud detection since these benefits are advantageous to the solution of real-world problems. This work aims to make the following new contributions:

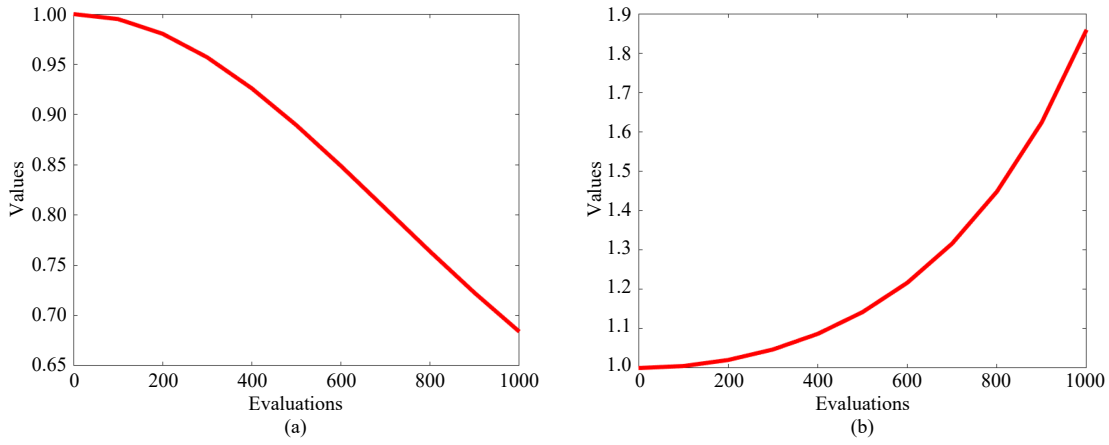
- 1) Proposing a simple and efficient dandelion algorithm in which all unnecessary mechanisms of DA are removed.
- 2) Developing an adaptive seeding radius strategy to reduce the number of DA's parameters.
- 3) Applying the proposed algorithm successfully to feature selection for credit card fraud detection.

The paper is organized as follows. In Section II, we provide a brief introduction of basic DA. The proposed algorithm is given in details in Section III. Section IV presents the experimental results. The application of SEDA is presented in Section V. Finally, Section VI draws the conclusion.

II. DANDELION ALGORITHM

DA is divided into four parts: initialization, normal sowing, mutation sowing, and selection strategy [12], [28]:

- 1) *Initialization*: The first generation of population is generated randomly in the search range of N dandelions.
- 2) *Normal Sowing*: DA solves a minimization problem where the number of seeds is calculated based on the fitness


 Fig. 1. Variation trend (a) $\gamma = 1$ and (b) $\gamma \neq 1$.

value, i.e., the smaller the fitness value, the more seeds are produced. Each dandelion produces dandelion seeds within a certain seeding radius that is dynamically adjusted.

3) *Mutation Sowing*: The best dandelion, i.e., the dandelion with the minimum fitness, jumps out of a local optimum by Levy mutation.

4) *Selection Strategy*: The best dandelion is always selected and retained for the next generation. Then $N - 1$ dandelions are selected from the rest according to a selection operator.

Normal sowing and selection strategies are repeated until the given or termination condition is satisfied.

In DA, the number of dandelion seeds is calculated as

$$N_i = \begin{cases} S_M \times \frac{f_M - f(x_i) + \varepsilon}{f_M - f_m + \varepsilon}, & N_i > S_m \\ S_m, & N_i \leq S_m \end{cases} \quad (1)$$

where S_M is used to control the maximum value of produced seeds, S_m is used to control the minimum number of produced seeds, f_M and f_m respectively the maximum and minimum fitness values of all dandelions, ε is machine epsilon, used to avoid the error of dividing by zero, and the fitness value of the i -th dandelion is expressed as $f(x_i)$.

Dandelions are divided into best dandelions and other dandelions according to their fitness value. Among them, the best dandelion is calculated as

$$R_b(t) = \begin{cases} U_b - L_b, & t = 1 \\ R_b(t-1) \times \alpha, & \gamma = 1 \\ R_b(t-1) \times \beta, & \gamma \neq 1 \end{cases} \quad (2)$$

where L_b and U_b is respectively lower bound and upper bound of the search space, t means the t -th generation, α and β are respectively withering factor and growth factor, and γ reflects the growth trend and is calculated as follows:

$$\gamma = \frac{f_b(t) + \varepsilon}{f_b(t-1) + \varepsilon}. \quad (3)$$

The sowing radius of other dandelions is calculated as:

$$R_i(t) = \begin{cases} U_b - L_b, & t = 1 \\ \delta \times R_i(t-1) + (\|x_b\|_\infty - \|x_i\|_\infty), & \text{otherwise} \end{cases} \quad (4)$$

where δ is a weight factor calculated as

$$\delta = 1 - \frac{T}{\hat{T}} \quad (5)$$

where \hat{T} is the maximum number of function evaluations, and T is the current number.

The best dandelion is sown with mutation operation, i.e.,

$$X'_b = X_b \times (1 + \text{Levy}()) \quad (6)$$

where $\text{Levy}()$ is a random number following the Levy distribution.

III. PROPOSED ALGORITHMS

A. A Self-Adapting and Efficient Dandelion Algorithm

In order to simplify DA's structure, a self-adapting and efficient dandelion algorithm (SEDA) is presented in this paper. In it, an adaptive seeding radius strategy is designed to further reduce the number of parameters when the sowing radius is calculated. The sowing radius for the core dandelion can be calculated as follows.

As for DA, dandelions are divided into two types: optimal and other dandelions, which are calculated via (2) and (3), respectively. They show that the optimal dandelion algorithm has two parameters when performing a seeding process: wilting and growth factors. To simplify DA's structure and reduce its parameter count, an adaptive dandelion algorithm keeps only one type of dandelion, i.e., the core dandelion. To simplify the structure of the algorithm and reduce parameter count, the adaptive dandelion algorithm keeps only one dandelion, i.e., the core dandelion, and removes the wilt and growth factors. The seeding radius of the core dandelion is calculated as.

$$R_b(t) = \begin{cases} U_b - L_b, & t = 1 \\ R_b(t-1) \times \frac{(1 + \beta^{-t^2/\hat{T}^2})}{2}, & \gamma = 1 \\ R_b(t-1) \times \frac{(1 + \beta^{t^2/\hat{T}^2})}{2}, & \gamma \neq 1. \end{cases} \quad (7)$$

From (7), it can be seen that the wilting and growth factors have been replaced by an adaptive radius strategy that can be updated adaptively by the number of function evaluations, and their trends are shown in Fig. 1.

As can be seen from Fig. 1, when $\gamma = 1$, the seeding radius becomes smaller as t increases, with a downward trend; when $\gamma \neq 1$, the seeding radius becomes larger as t increases, with an upward trend. They tend to meet the original intent of the underlying dandelion algorithm.

The proposed SEDA has one dandelion only in the initial population, and it can be considered as the best dandelion, which means that only the best dandelion generates the seeds and needs the calculation of its sowing radius. It is evident that SEDA is simpler than the basic DA and its variants. In SEDA, the sowing radius of the best dandelion is first calculated by (7), and then N_i seeds are generated with the sowing radius. If the new seeds are out of bounds, they are replaced by the random ones generated in the search space. Finally, the best one is selected from the population including the best one and all seeds by a greedy selection strategy. If the termination condition is not satisfied, the above steps are repeated.

In SEDA, all additional mechanisms (Gaussian variation, Levy variation, and seeding of common dandelions) have been removed as shown in Algorithm 1.

Algorithm 1 Self-Adapting and Efficient Dandelion Algorithm

Input: X_b and N_i // X_b is the current best dandelion, N_i is the number of seeds.

Output: X_i // Exporting the best dandelions.

```

1: repeat
2:   for  $i = 1$  to  $N_i$  do
3:     Calculate  $R_b$  by (7) // Calculate  $R_b$  using (7)
4:     for  $k = 1$  to  $d$  do
5:        $X_i^k = X_b^k + \text{rand}(0, R_b)$ 
6:       if  $X_i^k$  out of range then
7:         Generate a random location in the search space
8:       end if
9:     end for
10:  end for
11:  Select the best dandelion in the current population
12: until Satisfaction of termination conditions

```

To express the proposed algorithm more clearly, the flowchart of SEDA is given in Fig. 2. The computational complexity of Algorithm 1 is $O(\hat{T} \times M)$ where M is the total number of seeds, and \hat{T} is the maximum number of generations.

B. An Imbalance Classification Method Based on SEDA Feature Selection

Credit card fraud is an important issue and is quite costly for banks and card-related companies. Billions of dollars are lost each year due to credit card fraud. Due to confidentiality issues, there is a lack of research on analyzing real-world credit card data. Therefore, it is extremely important to detect whether a credit card is fraudulent. Credit card fraud detection is a classic imbalanced classification task [29], [30]. Feature selection is considered as an effective way to handle it [31]. In this paper, the proposed SEDA is applied to feature selection for solving it.

In SEDA, the dimension of each dandelion is set to the

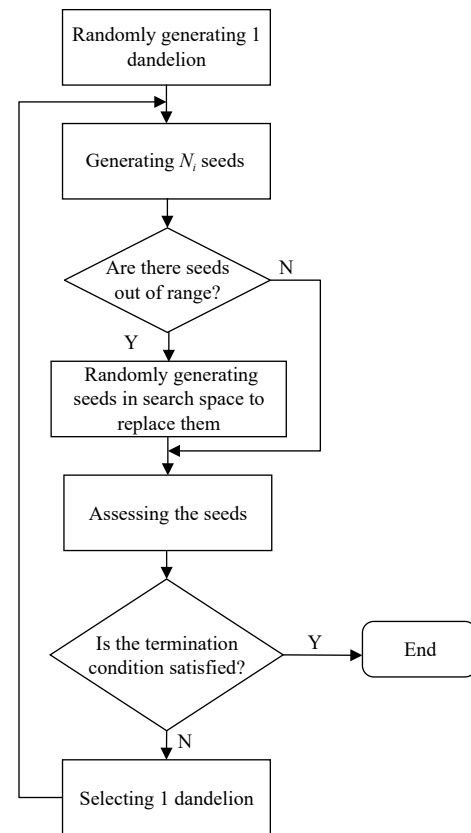


Fig. 2. Flowchart of SEDA.

number of features in a dataset. If the value of a dimension is greater than 0.5, it means that this dimension (feature) is selected. Otherwise, it is removed. For example, one dandelion (0.4, 0.1, 0.6, 0.7, 0.2) is to select the third and fourth feature from the dataset as the new dataset. In addition, the aims are to obtain a better classification performance and fewer features, and thus the objective function is designed as

$$f = \lambda \times (1 - G) + \mu \times N_F \quad (8)$$

where λ and μ are weight factors to trade-off the classification performance and number of features. They are set to 0.99 and 0.01 in this paper since the classification performance is more important than the number of features. G is the performance indicator for imbalanced classification (G -mean) [20]. N_F is the number of features. The smaller the value of f , the better the classification performance. Moreover, some of the state-of-the-art methods that are often used as comparisons, DA, MDA, DAPM_L, DAPM_B, DAPM_E, BA, and IPSO are selected to compare with SEDA. The maximum number of evaluations is set to 1000. It is noted the weighted extreme learning machine (WELM) [32] is selected as the base classifier and the reasons are: 1) it has fast learning speed; and 2) it is suitable for imbalanced classification problems.

G -mean and AUC (area under curve) are selected as the evaluation indicators. In the following, they are first conducted on 10 imbalanced classification datasets, and then three credit card fraud detection datasets.

In summary, the imbalance classification method based on

the adaptive efficient dandelion algorithm feature selection (SEDA-W) is shown in Algorithm 2.

Algorithm 2 Imbalanced Classification Algorithm Based on SEDA Feature Selection

Input: Training dataset, maximum number of evaluations.

Output: G -mean.

- 1: Initializing the parameters of SEDA and WELM
 - 2: Randomly initialize the population, calculate the fitness value of the population and evaluate the population according to (8)
 - 3: **repeat**
 - 4: Use (7) to generate seeds and evaluate the seeds
 - 5: Selection of individuals from existing populations for the next generation through a selection strategy
 - 6: **until** meet the stopping conditions
 - 7: Find the best dandelion and its location
 - 8: Get G -mean based on the best dandelion and its location
 - 9: **return**
-

In Algorithm 2, Lines 1 and 2 are the initialization phase that requires N function evaluations. Lines 4–6 require M function evaluations, where M is the total number of seeds. The computational complexity of WELM classification is $O(\hat{N}mL + \hat{N}^2L)$, where \hat{N} and m denote the number of training instances and the dimension of each training instance, respectively [33]. L denotes the number of hidden nodes in WELM. If the maximum number of generations are set to \hat{T} , the computational complexity of Algorithm 2 is $O(N(\hat{N}mL + \hat{N}^2L) + \hat{T}M(\hat{N}mL + \hat{N}^2L))$.

C. Convergence Analysis of SEDA-W

The convergence of SEDA-W is mathematically analyzed. We assume that X is a dandelion in SEDA-W's population. Since it has only one dandelion in the initial population, it can be considered as the best one.

Theorem 1: The evolution direction of SEDA-W is monotonically nonincreasing, i.e., $f(X(t+1)) \leq f(X(t))$, $\forall t \geq 1$.

Proof: Through the selection strategy, the best dandelion is always retained, and its fitness value is monotonically non-increasing. Thus, $f(X(t+1)) \leq f(X(t))$ is established. ■

Theorem 2: $\{X(t), t \in \mathbb{Z}^+\}$ of SEDA-W forms a homogeneous Markov chain.

Proof: $X(t)$ changes to $X(t+1)$, denoted as

$$X(t+1) = T(X(t)). \quad (9)$$

From (9), we can see that $X(t+1)$ is only related to the present state $X(t)$. Thus $X(t)$ of SEDA-W, $t > 0$, forms a Markov chain. Also, we can observe that T depends on $X(t)$, but not t [15]. ■

Theorem 3: $\{X(t), t \in \mathbb{Z}^+\}$ of SEDA-W converges to the global/local optimal solution set W^* with probability 1, i.e.,

$$\lim_{t \rightarrow \infty} P(X(t) \in W^*) = 1. \quad (10)$$

Proof: Based on Theorems 1 and 2, $\{X(t)\}$ is monotonically non-increasing homogeneous Markov chain, and the transition probability from $\{X(t-1)\}$ to $\{X(t)\}$ is [15]

$$P(X(t-1), X(t)) = \begin{cases} P(X(t) \in W^* | X(t-1) \in W^*) = 1 \\ P(X(t) \notin W^* | X(t-1) \in W^*) = 0 \\ P(X(t) \in W^* | X(t-1) \notin W^*) > 0 \\ P(X(t) \notin W^* | X(t-1) \notin W^*) > 0. \end{cases} \quad (11)$$

According to the total probability formula, we have

$$\begin{aligned} P(X(t) \in W^*) &= (1 - P(X(t-1) \in W^*)) \\ &\quad \times P(X(t) \in W^* | X(t-1) \notin W^*) \\ &\quad + P(X(t-1) \in W^*) \\ &\quad \times P(X(t) \in W^* | X(t-1) \in W^*). \end{aligned} \quad (12)$$

Let $\delta(t) = P(X(t) \in W^*)$, $u(t) = P(X(t) \in W^* | X(t-1) \notin W^*)$, (12) can be changed to

$$\delta(t) = (1 - \delta(t-1)) \times u(t) + \delta(t-1). \quad (13)$$

From (13), we have

$$\begin{aligned} 1 - \delta(t) &= (1 - \delta(t-1)) \times (1 - u(t)) \\ &= (1 - \delta(t-2)) \times (1 - u(t)) \times (1 - u(t-1)) \\ &= \\ &\vdots \\ &= (1 - \delta(0)) \times \prod_{k=1}^t (1 - u(k)). \end{aligned} \quad (14)$$

Taking the limit on both sides of (14), we obtain

$$\begin{aligned} \lim_{t \rightarrow \infty} \delta(t) &= 1 - \lim_{t \rightarrow \infty} \left((1 - \delta(0)) \times \prod_{k=1}^t (1 - u(k)) \right) \\ &= 1 - (1 - \delta(0)) \times \lim_{t \rightarrow \infty} \left(\prod_{k=1}^t (1 - u(k)) \right). \end{aligned} \quad (15)$$

From (15), it is easy to find that the result depends mainly on $u(k) = P(X(k) \in W^* | X(k-1) \notin W^*)$. If the function evaluations are given enough, SEDA-W could find the optimal solution, i.e., $0 < u(t) \leq 1$. Thus we have $\lim_{t \rightarrow \infty} (\prod_{k=1}^t (1 - u(k))) = 0$, and $\lim_{t \rightarrow \infty} \delta(t) = 1$, i.e., $\lim_{t \rightarrow \infty} P(X(t) \in W^*) = 1$. ■

IV. EXPERIMENTAL RESULTS AND THEIR ANALYSIS

A. Experimental Settings

1) *Benchmark Functions:* In order to evaluate the performance of the proposed algorithm, 28 benchmark functions from CEC2013 are used in our experiments. The CEC2013 test suite is listed in [34].

In this paper, each algorithm is run 51 times, and the dimension is set to 30, the function evaluations are set to 300 000. Note that no additional parameters need to be set in SEDA except the number of the seeds.

2) *Impact of the Number of Seeds:* On SEDA performance. The number of the seeds is set to 100, 200, 300, 400 and 500, respectively. The mean error and Friedman ranking are shown in Table I. The best mean error on each function is marked in bold. From Table I, it can be seen that SEDA performs the best among them when the number of the seeds is set to 300 in terms of Friedman ranking (2.5000). Thus, it is set to 300 in

TABLE I
MEAN ERROR AND FRIEDMAN RANKING OF SEDA FOR DIFFERENT NUMBERS OF SEEDS

Function	100	200	300	400	500
	\bar{e}	\bar{e}	\bar{e}	\bar{e}	\bar{e}
f_1	0.00E+00	0.00E+00	0.00E+00	1.94E-08	5.85E-06
f_2	3.11E+05	6.12E+05	9.92E+05	1.28E+06	1.73E+06
f_3	5.98E+07	5.69E+07	3.51E+07	3.93E+07	7.87E+07
f_4	2.09E-03	1.10E-03	1.11E-02	1.10E-01	6.64E-01
f_5	1.87E-04	1.72E-04	3.12E-04	9.63E-04	4.33E-03
f_6	2.32E+01	2.76E+01	2.76E+01	3.19E+01	2.95E+01
f_7	5.37E+01	5.18E+01	5.20E+01	4.74E+01	4.86E+01
f_8	2.09E+01	2.09E+01	2.09E+01	2.09E+01	2.09E+01
f_9	1.91E+01	1.92E+01	1.79E+01	1.83E+01	1.73E+01
f_{10}	2.98E-02	2.85E-02	2.36E-02	2.84E-02	3.05E-02
f_{11}	7.29E+01	7.25E+01	6.86E+01	7.35E+01	6.93E+01
f_{12}	1.06E+02	1.16E+02	1.14E+02	1.04E+02	9.81E+01
f_{13}	2.09E+02	2.05E+02	1.91E+02	1.97E+02	1.84E+02
f_{14}	2.21E+03	2.24E+03	2.14E+03	2.26E+03	2.16E+03
f_{15}	3.47E+03	3.48E+03	3.50E+03	3.28E+03	3.31E+03
f_{16}	2.86E-01	3.09E-01	2.16E-01	2.14E-01	2.49E-01
f_{17}	1.19E+02	1.09E+02	1.08E+02	1.07E+02	1.08E+02
f_{18}	1.56E+02	1.63E+02	1.60E+02	1.53E+02	1.57E+02
f_{19}	5.62E+00	5.69E+00	5.69E+00	5.22E+00	5.40E+00
f_{20}	1.21E+01	1.18E+01	1.18E+01	1.19E+01	1.16E+01
f_{21}	3.30E+02	3.28E+02	2.98E+02	3.07E+02	3.27E+02
f_{22}	2.63E+03	2.52E+03	2.39E+03	2.58E+03	2.27E+03
f_{23}	3.88E+03	3.90E+03	4.09E+03	3.75E+03	3.79E+03
f_{24}	2.55E+02	2.51E+02	2.55E+02	2.53E+02	2.48E+02
f_{25}	2.79E+02	2.79E+02	2.77E+02	2.74E+02	2.73E+02
f_{26}	2.06E+02	2.03E+02	2.00E+02	2.03E+02	2.03E+02
f_{27}	8.36E+02	8.14E+02	7.96E+02	7.87E+02	7.99E+02
f_{28}	3.40E+02	3.45E+02	2.84E+02	3.21E+02	3.11E+02
Friedman ranking	3.7142	3.5357	2.5000	2.6428	2.6071

the following experiment.

B. Comparison of SEDA With Other Methods

1) *Comparison of SEDA With DA and Its Variants*: In this section, the proposed SEDA is compared with the basic DA and its variants including MDA, DAPM_L, DAPM_B, and DAPM_E. Their parameter settings follow their corresponding references [12]. The mean error and Friedman rankings are shown in Table II. The best mean error and Friedman ranking are marked in bold.

From Table II, it can be seen that the six algorithms have the same performance on f_1 and f_8 , while SEDA has a better mean error than the other five algorithms on the other functions. Judging from the Friedman ranking, it can be concluded that the proposed SEDA is the best among them. The convergence curves of f_8 , f_9 , f_{13} , and f_{14} are shown in Fig. 3 while the other 24 convergence curves are in Supplementary File. From Fig. 3, it can be seen that our method can jump out of the local optimum in the process of running.

In addition, the Wilcoxon rank-sum test is conducted to verify the performance of all compared algorithms, in which “+” indicates that SEDA performs significantly better than its peers, “-” indicates SEDA performs worse than its peers, and “=” indicates SEDA and its peer have no insignificant difference. The results as presented at the bottom of Table II show that SEDA is significantly better than DA and its variants.

Finally, the average time consumption of six algorithms on 28 test functions is shown in Fig. 4. From it, it can be concluded that the proposed SEDA can achieve better performance with less time consumption.

Fig. 3 shows their shatis of DA, DAPM_L, DAPM_B, DAPM_E, IPSO, MDA, and SEDA on some benchmark functions.

2) *Comparison of Three Models With Other Swarm Intelligence Algorithms*: In order to further verify performance of the proposed SEDA, a comparison with ABC, IPSO, DE and CMA-ES is conducted on CEC2013, and the results of the four algorithms are taken from their references [35]–[38]. The comparison results of among ABC, DE, CMA-ES, IPSO, and

TABLE II
MEAN ERROR OF THE TEST FUNCTION AND FRIEDMAN RANKING

Function	DA	MDA	DAPM _L	DAPM _B	DAPM _E	SEDA
f_1	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
f_2	3.16E+05	4.57E+05	3.15E+05	3.13E+05	3.01E+05	9.92E+05
f_3	1.28E+08	1.16E+08	9.03E+07	1.13E+08	7.16E+07	3.51E+07
f_4	4.02E-01	2.13E-01	4.02E-01	3.34E-01	3.00E-01	1.11E-02
f_5	2.72E-04	3.10E-04	2.82E-04	2.59E-04	2.61E-04	3.12E-04
f_6	2.66E+01	2.69E+01	2.59E+01	2.59E+01	1.96E+01	2.76E+01
f_7	8.41E+01	8.77E+01	7.97E+01	8.50E+01	9.06E+01	5.20E+01
f_8	2.09E+01	2.09E+01	2.09E+01	2.09E+01	2.09E+01	2.09E+01
f_9	2.13E+01	2.27E+01	2.11E+01	2.18E+01	2.16E+01	1.79E+01
f_{10}	2.95E-02	4.36E-02	2.82E-02	3.10E-02	3.23E-02	2.36E-02
f_{11}	8.76E+01	8.34E+01	9.40E+01	8.87E+01	9.18E+01	6.86E+01
f_{12}	1.35E+02	1.38E+02	1.25E+02	1.24E+02	1.37E+02	1.14E+02
f_{13}	2.30E+02	2.36E+02	2.25E+02	2.24E+02	2.26E+02	1.91E+02
f_{14}	2.72E+03	2.83E+03	2.76E+03	2.72E+03	2.73E+03	2.14E+03
f_{15}	3.74E+03	3.62E+03	3.46E+03	3.68E+03	3.67E+03	3.50E+03
f_{16}	3.41E-01	5.21E-01	3.16E-01	3.41E-01	3.34E-01	2.16E-01
f_{17}	1.35E+02	1.33E+02	1.31E+02	1.30E+02	1.37E+02	1.08E+02
f_{18}	1.89E+02	1.88E+02	1.76E+02	1.80E+02	1.85E+02	1.60E+02
f_{19}	6.51E+00	6.17E+00	6.06E+00	5.93E+00	6.11E+00	5.69E+00
f_{20}	1.22E+01	1.20E+01	1.20E+01	1.20E+01	1.20E+01	1.18E+01
f_{21}	3.39E+02	3.30E+02	3.20E+02	3.45E+02	3.47E+02	2.98E+02
f_{22}	3.17E+03	3.20E+03	3.15E+03	3.18E+03	3.10E+03	2.39E+03
f_{23}	4.08E+03	4.07E+03	4.14E+03	4.30E+03	4.17E+03	4.09E+03
f_{24}	2.62E+02	2.68E+02	2.65E+02	2.65E+02	2.65E+02	2.55E+02
f_{25}	2.86E+02	2.90E+02	2.86E+02	2.89E+02	2.87E+02	2.77E+02
f_{26}	2.06E+02	2.10E+02	2.03E+02	2.06E+02	2.06E+02	2.00E+02
f_{27}	8.96E+02	9.15E+02	8.95E+02	8.76E+02	8.99E+02	7.96E+02
f_{28}	4.05E+02	3.19E+02	3.68E+02	3.41E+02	3.95E+02	2.84E+02
Friedman ranking	4.1964	4.625	3.0892	3.4821	3.875	1.7321
+/-	17/8/3	19/7/2	17/9/2	16/9/3	18/7/3	N/A

SEDA are presented in Table III. The best result is marked in bold.

As shown in Table III, CMA-ES performs extremely well on unimodal functions but suffers from premature convergence on some complex functions. ABC performs best on 10 functions. Judging from the Friedman ranking, it can be seen that the SEDA is the best among all the compared algorithms, which indicates that SEDA is more stable on the test function.

C. Comparison on 13 Imbalanced Datasets

In order to verify the performance of SEDA, a comparison among DA, MDA, DAPM_L, DAPM_B, DAPM_E, BA, and IPSO with WELM are conducted on 13 imbalanced classification datasets in this section, as shown in Table IV. The datasets can be found in UCI and KEEL database, where column “Abbr.” lists the assigned code of datasets, “#N” is the number of samples, “#mi/#Ma” represents the number of minority and majority samples, “#A” is the number of features in a

sample, and “#R” is the ratio of the number of minority samples to the number of majority samples in a dataset.

In this simulation, the regularization coefficient and bandwidth of the radial basis function kernel in WELM are set to 1 and 100, respectively. Each algorithm runs 10 times. The results of G -mean [20] and AUC [39] are shown in Tables V and VI.

In addition, AUC and G -mean are evaluation indexes to measure the quality of dichotomies. AUC is defined as the area under the ROC (receiver operating characteristic) curve. F_1 represents the harmonic average of *Recall* and *Precision*. G -mean is the geometric average of the prediction accuracy of positive and negative samples.

$$Precision = \frac{TP}{TP + FP} \quad (16)$$

$$Recall = \frac{TP}{TP + FN} \quad (17)$$

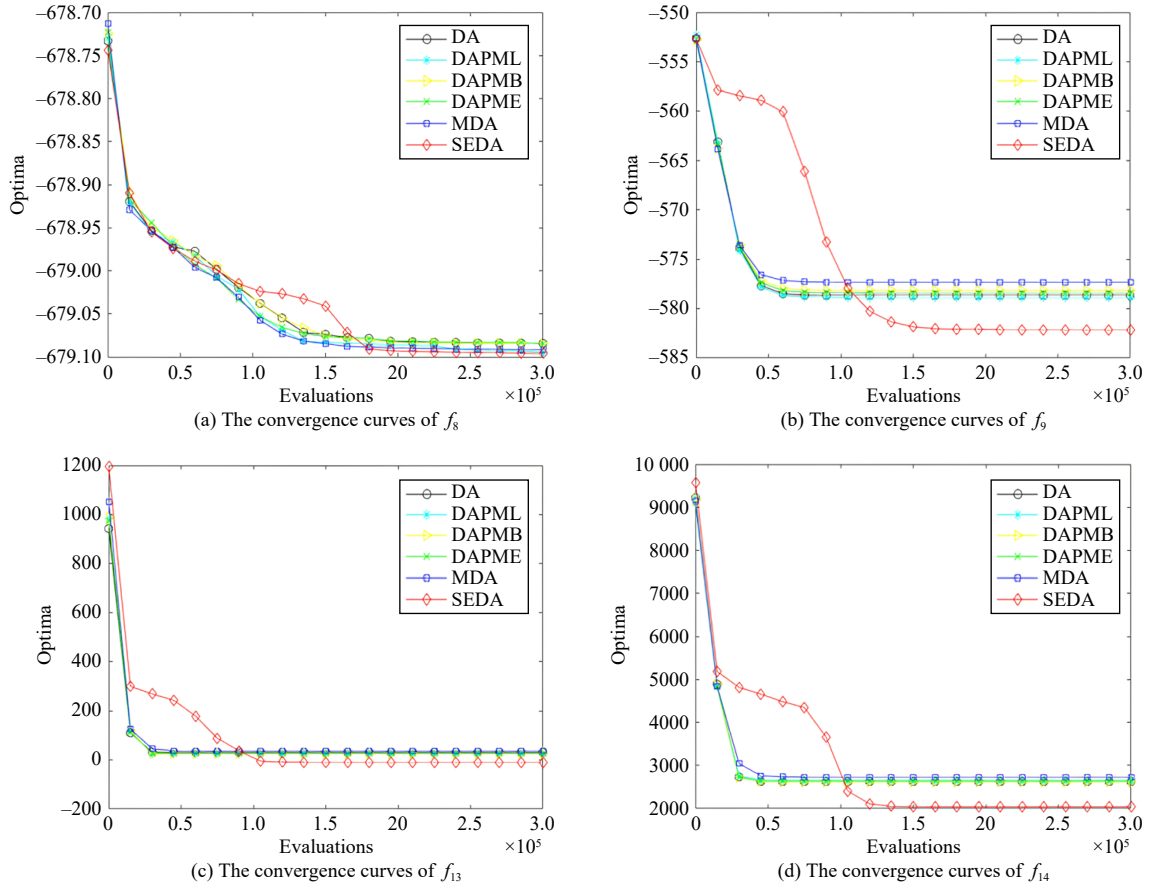


Fig. 3. Convergence curves of DA, DAPML, DAPMB, DAPME, MDA, and SEDA on some benchmark functions.

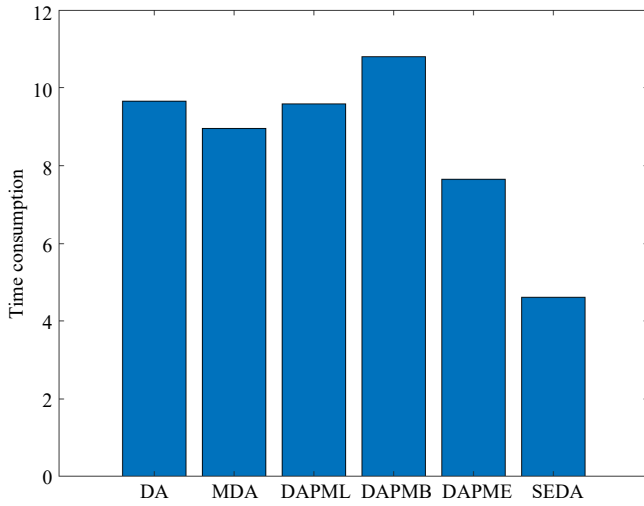


Fig. 4. Average time cost of 6 algorithms on 28 tested functions.

$$F_1 = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (18)$$

$$G\text{-mean} = \sqrt{\frac{T_P}{T_P + F_N} \times \frac{T_N}{T_N + F_P}} \quad (19)$$

where T_P , T_N , F_P , and F_N are respectively the number of true positive cases, true negative cases, false positive cases, and false negative cases.

From Table V, it can be seen that SEDA performs the best

among them on S_1 , and S_4 – S_{13} , and it is a bit worse than IPSO on S_2 and S_3 . DAPML and SEDA have the same performance on S_{13} . The other algorithms perform poorly in the remaining datasets. In addition, the results of the T-test, as shown at the bottom of Table V, indicate that SEDA is significantly better than its peers in terms of G-mean values. From Table VI, it can be observed that SEDA is the best on 12 datasets besides S_2 in terms of AUC.

Based on the results of G -mean and AUC, we can conclude that SEDA can pick out a better feature subset from all features to get better classification performance than its peers.

D. Comparison With Three Datasets With Large Imbalance Ratio

In order to investigate the performance of the proposed SEDA on datasets with large imbalance ratios, we have selected three datasets which are from [40], as shown in Table VII. Dataset S_{16} contains a total of 284 807 samples, with only 492 samples belonging to the minority class, accounting for only 0.172% of the total. The results of G -mean and AUC are presented in Tables VIII and IX, and the best results on each dataset are marked in bold. As can be seen from Tables VIII and IX, on all three larger imbalance ratio datasets, our proposed method outperforms the other methods on G -mean, and the AUC value of S_{15} is slightly lower than that of MDA-W.

E. Comparison via Three CCFD Datasets

Three real-world transaction datasets are selected for this

TABLE III
MEAN ERROR AND FRIEDMAN RANKING OF THE FIVE ALGORITHMS

Function	ABC	DE	CMA-ES	IPSO	SEDA
f_1	0.00E+00	1.89E-03	0.00E+00	0.00E+00	0.00E+00
f_2	6.20E+06	5.52E+04	0.00E+00	3.38E+05	9.92E+05
f_3	5.74E+08	2.16E+06	1.41E+01	2.88E+08	3.51E+07
f_4	8.75E+04	1.32E-01	0.00E+00	3.86E+04	1.11E-02
f_5	0.00E+00	2.48E-03	0.00E+00	5.42E-04	3.12E-04
f_6	1.46E+01	7.82E+00	7.82E-02	3.79E+01	2.76E+01
f_7	1.25E+02	4.89E+01	1.91E+01	8.79E+01	5.20E+01
f_8	2.09E+01	2.09E+01	2.14E+01	2.09E+01	2.09E+01
f_9	3.01E+01	1.59E+01	4.81E+01	2.88E+01	1.79E+01
f_{10}	2.27E-01	3.42E-02	1.78E-02	3.40E-01	2.36E-02
f_{11}	0.00E+00	7.88E+01	4.00E+02	1.05E+02	6.86E+01
f_{12}	3.19E+02	8.14E+01	9.42E+02	1.04E+02	1.14E+02
f_{13}	3.29E+02	1.61E+02	1.08E+03	1.94E+02	1.91E+02
f_{14}	3.58E-01	2.38E+03	4.94E+03	3.99E+03	2.14E+03
f_{15}	3.88E+03	5.19E+03	5.02E+03	3.81E+03	3.50E+03
f_{16}	1.07E+00	1.97E+00	5.42E-02	1.31E+00	2.16E-01
f_{17}	3.04E+01	9.29E+01	7.44E+02	1.16E+02	1.08E+02
f_{18}	3.04E+02	2.34E+02	5.17E+02	1.21E+02	1.60E+02
f_{19}	2.62E-01	4.51E+00	3.54E+00	9.51E+00	5.69E+00
f_{20}	1.44E+01	1.43E+01	1.49E+01	1.35E+01	1.18E+01
f_{21}	1.65E+02	3.20E+02	3.44E+02	3.09E+02	2.98E+02
f_{22}	2.41E+01	1.72E+03	7.97E+03	4.30E+03	2.39E+03
f_{23}	4.95E+03	5.28E+03	6.95E+03	4.83E+03	4.09E+03
f_{24}	2.90E+02	2.47E+02	6.62E+02	2.67E+02	2.55E+02
f_{25}	3.06E+02	2.89E+02	4.41E+02	2.99E+02	2.77E+02
f_{26}	2.01E+02	2.52E+02	3.29E+02	2.86E+02	2.00E+02
f_{27}	4.16E+02	7.64E+02	5.39E+02	1.00E+03	7.96E+02
f_{28}	2.58E+02	4.02E+02	4.78E+03	4.01E+02	2.84E+02
Friedman ranking	2.875	2.8393	3.5357	3.3929	2.3571

TABLE IV
PUBLIC DATA

Abbr.	Dataset	#N	#mi/Ma	#A	#R
S ₁	German	1000	300/700	20	2.3
S ₂	Phoneme	5404	1586/3818	5	2.4
S ₃	Haberman	306	81/225	3	2.8
S ₄	Cmc	1473	333/1140	9	3.4
S ₅	Balance	625	49/576	4	11.8
S ₆	Vehicle	846	212/634	18	3.0
S ₇	Glass2	214	17/197	9	11.6
S ₈	Boston housing	506	106/400	13	3.8
S ₉	Hepatitis	155	32/123	19	3.8
S ₁₀	Nursery	12 960	328/12 632	8	38.5
S ₁₁	Wpbc	198	47/151	33	3.2
S ₁₂	Satimage	6435	626/5809	36	9.3
S ₁₃	Abalone	4177	36/4141	9	115.0

TABLE V
G-MEAN

Dataset	BA-W	DA-W	DAPM _L -W	DAPM _B -W	DAPM _E -W	IPSO-W	MDA-W	SEDA-W
S ₁	0.6875	0.6797	0.6808	0.6666	0.6523	0.7170	0.6853	0.7451
S ₂	0.7243	0.7427	0.7429	0.7430	0.7429	0.7519	0.7428	0.7478
S ₃	0.5843	0.6013	0.6039	0.6018	0.6060	0.6395	0.6013	0.6361
S ₄	0.6374	0.6601	0.6614	0.6603	0.6604	0.6778	0.6614	0.6922
S ₅	0.3973	0.5169	0.5179	0.5205	0.5179	0.5429	0.5175	0.5469
S ₆	0.6529	0.6692	0.6641	0.6583	0.6581	0.6737	0.6667	0.7120
S ₇	0.5758	0.7291	0.7291	0.7219	0.7220	0.6270	0.7291	0.7597
S ₈	0.5724	0.6297	0.6306	0.6309	0.6314	0.6136	0.6324	0.7015
S ₉	0.7381	0.8984	0.8905	0.8921	0.8343	0.8579	0.8764	0.9537
S ₁₀	0.8459	0.8348	0.8445	0.8343	0.8412	0.8580	0.8343	0.8623
S ₁₁	0.6651	0.7636	0.7403	0.6972	0.6963	0.7109	0.7625	0.7921
S ₁₂	0.6017	0.7152	0.6921	0.6821	0.6800	0.6431	0.6753	0.7240
S ₁₃	0.7083	0.7971	0.7981	0.7862	0.7864	0.7731	0.7917	0.7981
<i>t</i> -test (+/-/-)	13/0/0	11/2/0	11/2/0	12/1/0	11/2/0	9/1/3	11/2/0	N/A

TABLE VI
AUC

Dataset	BA-W	DA-W	DAPM _L -W	DAPM _B -W	DAPM _E -W	IPSO-W	MDA-W	SEDA-W
S ₁	0.6910	0.6864	0.6912	0.6710	0.6683	0.7250	0.6921	0.7514
S ₂	0.7262	0.7438	0.7437	0.7439	0.7437	0.7526	0.7436	0.7502
S ₃	0.6229	0.6233	0.6260	0.6219	0.6334	0.6573	0.6233	0.6608
S ₄	0.6630	0.6637	0.6640	0.6621	0.6642	0.6828	0.6640	0.6983
S ₅	0.4749	0.5272	0.5278	0.5338	0.5277	0.5542	0.5262	0.5559
S ₆	0.6580	0.6764	0.6708	0.6663	0.6658	0.6756	0.6716	0.7151
S ₇	0.6599	0.7686	0.7696	0.7616	0.7636	0.6999	0.7684	0.7911
S ₈	0.6010	0.6501	0.6506	0.6592	0.6583	0.6457	0.6526	0.7348
S ₉	0.7500	0.9054	0.8971	0.8769	0.8452	0.8660	0.8804	0.9554
S ₁₀	0.8580	0.8501	0.8555	0.8692	0.8556	0.8682	0.8651	0.8718
S ₁₁	0.6779	0.7819	0.7588	0.7119	0.7121	0.7269	0.7721	0.7994
S ₁₂	0.6767	0.7317	0.7185	0.7195	0.7104	0.7042	0.7127	0.7534
S ₁₃	0.7267	0.8038	0.8041	0.7933	0.7936	0.7906	0.7977	0.8041
<i>t</i> -test (+/-/-)	13/0/0	11/2/0	11/2/0	10/3/0	11/2/0	9/3/1	10/3/0	N/A

experiment to validate the performance of SEDA, they are LoanPrediction¹, Creditcardsvpresent² and Default of Credit Card Clients³. For convenience, they are abbreviated as D₁–D₃, respectively, which list in Table X.

To reflect the classification performance comprehensively, in addition to *G*-mean and AUC, three evaluation metrics, *Precision*, *Recall*, and *F*₁, are added to measure the performance of each algorithm, and the experimental results are shown in Tables XI–XIII.

The experimental results are analysed as follows.

1) IPSO has the best *Precision* on the dataset D₁; SEDA has better results than other algorithms on *Recall*, *F*₁, AUC and *G*-mean. On the key metric *G*-mean, SEDA beats by 4.50% and

9.06% the second best method IPSO-W and the worst one BA-W, respectively.

2) On dataset D₂, SEDA has the best results on all five evaluation metrics. On the key metric *G*-mean, SEDA outforms by 0.85% and 12.08% the second best method DA-W and the worst one DAPM_E-W, respectively.

3) IPSO has the best *Precision* and *F*₁ on dataset D₃; SEDA has the best results on *Recall*, AUC and *G*-mean. On the key metric *G*-mean, SEDA performs better 1.44% and 9.41% than the second best method IPSO-W and worst one DAPM_E-W, respectively.

From the analysis of the above experimental results, it can be seen that the method proposed in this paper outperforms the compared methods in most of the indicators, especially in the key indicator *G*-mean, which is better than all other methods.

¹ https://github.com/Paliking/ML_examples/blob/master/LoanPrediction/train_u6lujuX_CVtuZ9i.csv

² <https://github.com/gksj7/creditcardsvpresent>

³ <http://archive.ics.uci.edu/ml/datasets/default+of+credit+card+clients>

TABLE VII
THE DATASETS OF LARGE IMBALANCE RATIO

Abbr.	Dataset	#N	#mi/Ma	#A	#R
S ₁₄	Statlog (Shuttle)	46 036	450/44 586	8	101.3
S ₁₅	Default of credit card clients	23 430	66/23 364	22	354.0
S ₁₆	Credit card fraud detection	284 807	492/284 315	30	577.9

TABLE VIII
G-MEAN OF THE LARGE IMBALANCE RATIO DATASETS

Dataset	BA-W	DA-W	DAPM _L -W	DAPM _B -W	DAPM _E -W	IPSO-W	MDA-W	SEDA-W
S ₁₄	0.8237	0.9100	0.9102	0.6114	0.9102	0.9180	0.9102	0.9285
S ₁₅	0.5004	0.5746	0.6307	0.5707	0.6306	0.5229	0.6308	0.6319
S ₁₆	0.5002	0.5746	0.5787	0.5652	0.5446	0.5819	0.5380	0.5942
<i>t</i> -test (+/ = /-)	3/0/0	3/0/0	2/1/0	3/0/0	3/0/0	2/1/0	3/0/0	N/A

TABLE IX
AUC OF THE LARGE IMBALANCE RATIO DATASETS

Dataset	BA-W	DA-W	DAPM _L -W	DAPM _B -W	DAPM _E -W	IPSO-W	MDA-W	SEDA-W
S ₁₄	0.8696	0.9148	0.9150	0.7635	0.9150	0.9224	0.9150	0.9318
S ₁₅	0.5035	0.5864	0.6325	0.5726	0.6329	0.5756	0.6387	0.6385
S ₁₆	0.5009	0.5864	0.5898	0.5726	0.5555	0.5862	0.5481	0.5943
<i>t</i> -test (+/ = /-)	3/0/0	3/0/0	2/1/0	3/0/0	3/0/0	2/1/0	1/1/1	N/A

TABLE X
PUBLIC DATA

Abbr.	Dataset	#N	#mi/Ma	#A	#R
D ₁	LoanPrediction	614	192/422	11	0.45
D ₂	Default of credit card clients	30 000	6636/23 364	22	0.28
D ₃	CreditCardCSVPresent	3075	448/2627	9	0.17

TABLE XI
COMPARISON OF RESULTS FOR D₁

Method	Precision	Recall	F_1	AUC	G-mean
BA-W	0.7768	0.4539	0.5643	0.6915	0.6457
DA-W	0.7713	0.5408	0.5932	0.7039	0.6618
DAPM _L -W	0.7713	0.5408	0.5932	0.7039	0.6618
DAPM _B -W	0.7519	0.5462	0.5907	0.7012	0.6602
DAPM _E -W	0.7521	0.5461	0.5898	0.6994	0.6608
IPSO-W	0.8236	0.4945	0.6026	0.7153	0.6739
MDA-W	0.7713	0.5408	0.5932	0.7039	0.6618
SEDA-W	0.7179	0.5989	0.6320	0.7210	0.7042

Finally, we examine the features selected by each algorithm with the largest *G*-mean values on the datasets D₁–D₃, as shown in Figs. 5–7. The red parts are selected and white one are unselected. As can be seen in Fig. 5, in D₁, the feature in column 10 is selected by all the algorithms in the experiment, and therefore this feature contributes more to the differentiation of categories than the rest, while the features in columns 1 and 7 contributed less to the differentiation of categories. As can be seen in Fig. 6, in D₂, the features in column 5 con-

tribute more to distinguishing categories than the rest, while the features in columns 1, 4, 8, 11, 16, 17 and 19 contribute less to distinguishing categories. As can be seen in Fig. 7, in D₃, the feature in column 5 contributes more to distinguishing categories than the rest, while the feature in column 3 contributes less to distinguishing categories.

V. CONCLUSION

This paper has presented a simple and efficient dandelion

TABLE XII
COMPARISON OF RESULTS FOR D_2

Method	Precision	Recall	F_1	AUC	G-mean
BA-W	0.5265	0.4985	0.5016	0.6681	0.6373
DA-W	0.5537	0.5458	0.5481	0.6983	0.6799
DAPM _L -W	0.5547	0.5207	0.5346	0.6904	0.6656
DAPM _B -W	0.5988	0.4485	0.5129	0.6527	0.6118
DAPM _E -W	0.5608	0.4497	0.4822	0.6539	0.6129
IPSO-W	0.5393	0.4853	0.5090	0.6722	0.6439
MDA-W	0.5032	0.5367	0.4978	0.6622	0.6255
SEDA-W	0.5666	0.5502	0.5571	0.7039	0.6857

TABLE XIII
COMPARISON OF RESULTS FOR D_3

Method	Precision	Recall	F_1	AUC	G-mean
BA-W	0.6580	0.7102	0.6426	0.8089	0.7962
DA-W	0.5025	0.7346	0.5951	0.8043	0.8007
DAPM _L -W	0.5026	0.7347	0.5969	0.8053	0.8018
DAPM _B -W	0.5015	0.7343	0.5960	0.8032	0.7908
DAPM _E -W	0.4818	0.7213	0.5766	0.7936	0.7899
IPSO-W	0.7155	0.7928	0.7250	0.8583	0.8519
MDA-W	0.5025	0.7346	0.5951	0.8043	0.8007
SEDA-W	0.6093	0.8440	0.6885	0.8674	0.8642

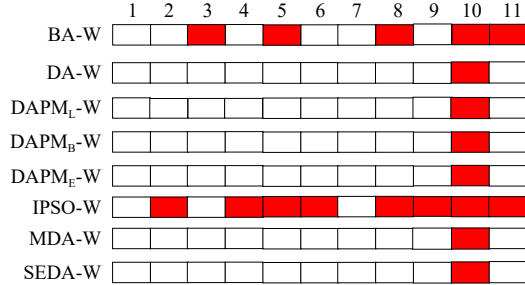


Fig. 5. D_1 selected features.

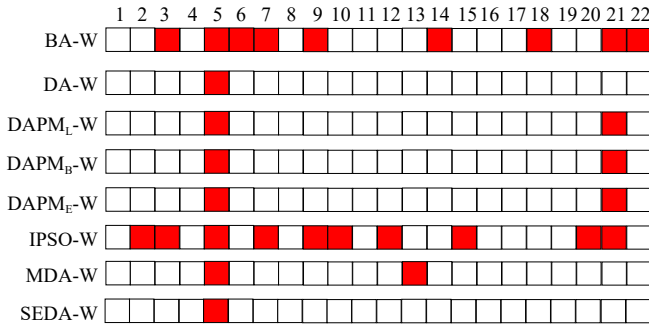


Fig. 6. D_2 selected features.

algorithm (SEDA) by removing all unnecessary mechanisms of basic DA, an adaptive seeding radius strategy is designed to further reduce the number of DA's parameters. Experimental results show that our proposed algorithm outperforms the

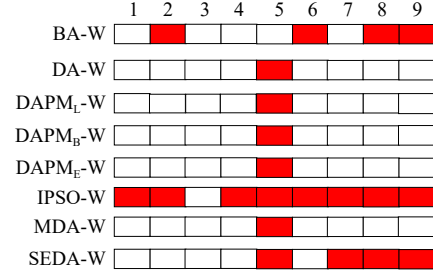


Fig. 7. D_3 selected features.

basic DA and its variants, as well as other swarm intelligence algorithms on CEC2013 benchmark functions. In addition, it is found that the proposed algorithm requires less time consumption than its compared peers. Note that SEDA has been combined with WELM for the classification problem of imbalanced dataset. The experiments are conducted on 13 public datasets, and 3 datasets are large ratio datasets. Experimental results show the superiority of our proposed method. Finally, it is applied to feature selection for credit card fraud detection, and the results show its effectiveness. All experimental data in this paper can be obtained from <https://github.com/bbxzyzh/Experimental-data>.

Our future work aims to analyze the other optimization technologies [41]–[49] and combine them with SEDA to further improve its performance. More real-world datasets [42], [50]–[52] should be used to test the proposed algorithm and its peers. In addition, in practice, the credit card data may be incomplete due to physical constraints. Hence, how to deal

with incomplete data efficiently should be the focus of our future work [53]–[55].

REFERENCES

- [1] S. Katoch, S. S. Chauhan, and V. Kumar, “A review on genetic algorithm: Past, present, and future,” *Multimed. Tools Appl.*, vol. 80, no. 5, pp. 8091–8126, Feb. 2021.
- [2] Ş. Öztürk, R. Ahmad, and N. Akhtar, “Variants of artificial bee colony algorithm and its applications in medical image processing,” *Appl. Soft Comput.*, vol. 97, p. 106799, Dec. 2020.
- [3] W. Deng, J. Xu, H. Zhao, and Y. Song, “A novel gate resource allocation method using improved PSO-based QEA,” *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 3, pp. 1737–1745, Mar. 2022.
- [4] E. B. Tirkolaee, A. Goli, and G. W. Weber, “Fuzzy mathematical programming and self-adaptive artificial fish swarm algorithm for just-in-time energy-aware flow shop scheduling problem with outsourcing option,” *IEEE Trans. Fuzzy Syst.*, vol. 28, no. 11, pp. 2772–2783, Nov. 2020.
- [5] J. R. Albert, A. Sharma, B. Rajani, A. Mishra, A. Saxena, C. Nandagopal, and S. Mewada, “Investigation on load harmonic reduction through solar-power utilization in intermittent SSFI using particle swarm, genetic, and modified firefly optimization algorithms,” *J. Intell. Fuzzy Syst.*, vol. 42, no. 4, pp. 4117–4133, Mar. 2022.
- [6] R. Rajabioun, “Cuckoo optimization algorithm,” *Appl. Soft Comput.*, vol. 11, no. 8, pp. 5508–5518, Dec. 2011.
- [7] J. Bi, H. Yuan, J. Zhai, M. Zhou, and H. V. Poor, “Self-adaptive bat algorithm with genetic operations,” *IEEE/CAA J. Autom. Sinica*, vol. 9, no. 7, pp. 1284–1294, Jul. 2022.
- [8] S. Mirjalili, S. Saremi, S. M. Mirjalili, and L. D. S. Coelho, “Multi-objective grey wolf optimizer: A novel algorithm for multi-criterion optimization,” *Expert Syst. Appl.*, vol. 47, pp. 106–119, Apr. 2016.
- [9] N. Hansen, S. D. Müller, and P. Koumoutsakos, “Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (CMA-ES),” *Evol. Comput.*, vol. 11, no. 1, pp. 1–18, Mar. 2003.
- [10] S. Han, K. Zhu, M. Zhou, X. Liu, H. Liu, Y. Al-Turki, and A. Abusorrah, “A novel multiobjective fireworks algorithm and its applications to imbalanced distance minimization problems,” *IEEE/CAA J. Autom. Sinica*, vol. 9, no. 8, pp. 1476–1489, Aug. 2022.
- [11] D. H. Wolpert and W. G. Macready, “No free lunch theorems for optimization,” *IEEE Trans. Evol. Comput.*, vol. 1, no. 1, pp. 67–82, Apr. 1997.
- [12] H. Zhu, G. Liu, M. Zhou, Y. Xie, and Q. Kang, “Dandelion algorithm with probability-based mutation,” *IEEE Access*, vol. 7, pp. 97974–97985, Jul. 2019.
- [13] C. Gong, S. Han, X. Li, L. Zhao, and X. Liu, “A new dandelion algorithm and optimization for extreme learning machine,” *J. Exp. Theor. Artif. Intell.*, vol. 30, no. 1, pp. 39–52, 2018.
- [14] S. Han and K. Zhu, “Fusion with distance-aware selection strategy for dandelion algorithm,” *Knowl.-Based Syst.*, vol. 205, p. 106282, Oct. 2020.
- [15] S. Han, K. Zhu, and R. Wang, “Improvement of evolution process of dandelion algorithm with extreme learning machine for global optimization problems,” *Expert Syst. Appl.*, vol. 163, p. 113803, Jan. 2021.
- [16] G.-B. Huang, Q.-Y. Zhu, and C. K. Siew, “Extreme learning machine: A new learning scheme of feedforward neural networks,” in *Proc. IEEE Int. Joint Conf. Neural Networks*, Budapest, Hungary, 2004, pp. 985–990.
- [17] S. Han, K. Zhu, and M. Zhou, “Competition-driven dandelion algorithms with historical information feedback,” *IEEE Trans. Syst. Man Cybern. Syst.*, vol. 52, no. 2, pp. 966–979, Feb. 2022.
- [18] J. Li, J. Q. Zhang, C. J. Jiang, and M. Zhou, “Composite particle swarm optimizer with historical memory for function optimization,” *IEEE Trans. Cybern.*, vol. 45, no. 10, pp. 2350–2363, Oct. 2015.
- [19] X. Liu and X. Qin, “A probability-based core dandelion guided dandelion algorithm and application to traffic flow prediction,” *Eng. Appl. Artif. Intell.*, vol. 96, p. 103922, Nov. 2020.
- [20] H. Zhu, G. Liu, M. Zhou, Y. Xie, A. Abusorrah, and Q. Kang, “Optimizing weighted extreme learning machines for imbalanced classification and application to credit card fraud detection,” *Neurocomputing*, vol. 407, pp. 50–62, Sept. 2020.
- [21] A. H. Hosseini and V. Baradaran, “Detecting communities of workforces for the multi-skill resource-constrained project scheduling problem: A dandelion solution approach,” *J. Ind. Syst. Eng.*, vol. 12, no. Special issue on Project Management and Control, pp. 72–99, Jan. 2019.
- [22] V. Bolón-Canedo, N. Sánchez-Marroño, A. Alonso-Betanzos, J. M. Benítez, and F. Herrera, “A review of microarray datasets and applied feature selection methods,” *Inf. Sci.*, vol. 282, pp. 111–135, Oct. 2014.
- [23] Z. Sun, G. Bebis, and R. Miller, “Object detection using feature subset selection,” *Pattern Recognit.*, vol. 37, no. 11, pp. 2165–2176, Nov. 2004.
- [24] B. Xue, M. Zhang, and W. N. Browne, “Particle swarm optimisation for feature selection in classification: Novel initialisation and updating mechanisms,” *Appl. Soft Comput.*, vol. 18, pp. 261–276, May 2014.
- [25] M. K. Masood, Y. C. Soh, and C. Jiang, “Occupancy estimation from environmental parameters using wrapper and hybrid feature selection,” *Appl. Soft Comput.*, vol. 60, pp. 482–494, Nov. 2017.
- [26] Y. Xue, B. Xue, and M. Zhang, “Self-adaptive particle swarm optimization for large-scale feature selection in classification,” *ACM Trans. Knowl. Discovery Data*, vol. 13, no. 5, p. 50, Oct. 2019.
- [27] R. Sagban, H. A. Marhoon, and R. Alubady, “Hybrid bat-ant colony optimization algorithm for rule-based feature selection in health care,” *Int. J. Electr. Comput. Eng.*, vol. 10, no. 6, pp. 6655–6663, Dec. 2020.
- [28] X. Li, S. Han, L. Zhao, C. Gong, and X. Liu, “New dandelion algorithm optimizes extreme learning machine for biomedical classification problems,” *Comput. Intell. Neurosci.*, vol. 2017, p. 4523754, Sept. 2017.
- [29] Q. Kang, L. Shi, M. Zhou, X. S. Wang, Q. D. Wu, and Z. Wei, “A distance-based weighted undersampling scheme for support vector machines and its application to imbalanced classification,” *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 9, pp. 4152–4165, Sept. 2018.
- [30] L. Zheng, G. Liu, C. Yan, and C. Jiang, “Transaction fraud detection based on total order relation and behavior diversity,” *IEEE Trans. Comput. Soc. Syst.*, vol. 5, no. 3, pp. 796–806, Sept. 2018.
- [31] H. Liu, M. Zhou, and Q. Liu, “An embedded feature selection method for imbalanced data classification,” *IEEE/CAA J. Autom. Sinica*, vol. 6, no. 3, pp. 703–715, May 2019.
- [32] W. Zong, G.-B. Huang, and Y. Chen, “Weighted extreme learning machine for imbalance learning,” *Neurocomputing*, vol. 101, pp. 229–242, Feb. 2013.
- [33] H. Yu, C. Sun, X. Yang, S. Zheng, Q. Wang, and X. Xi, “LW-ELM: A fast and flexible cost-sensitive learning framework for classifying imbalanced data,” *IEEE Access*, vol. 6, pp. 28488–28500, May 2018.
- [34] J. J. Liang, B. Y. Qu, P. N. Suganthan, and A. G. Hernández-Díaz, “Problem definitions and evaluation criteria for the CEC 2013 special session on real-parameter optimization,” *Comput. Intell. Lab., Zhengzhou Univ., Zhengzhou, China and Nanyang Technol. Univ., Singapore*, Tech. Rep., Jan. 2013.
- [35] D. Karaboga and B. Basturk, “A powerful and efficient algorithm for numerical function optimization: Artificial bee colony (ABC) algorithm,” *J. Global Optim.*, vol. 39, no. 3, pp. 459–471, Nov. 2007.
- [36] M. Zambrano-Bigiarini, M. Clerc, and R. Rojas, “Standard particle swarm optimisation 2011 at CEC-2013: A baseline for future PSO improvements,” in *Proc. IEEE Congr. Evolutionary Computation*, Cancun, Mexico, 2013, pp. 2337–2344.
- [37] R. Storn and K. Price, “Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces,” *J. Global Optim.*, vol. 11, no. 4, pp. 341–359, Dec. 1997.
- [38] N. Hansen and A. Ostermeier, “Adapting arbitrary normal mutation distributions in evolution strategies: The covariance matrix adaptation,” in *Proc. IEEE Int. Conf. Evolutionary Computation*, Nagoya, Japan, 1996, pp. 312–317.
- [39] J. M. Lobo, A. Jiménez-Valverde, and R. Real, “AUC: A misleading measure of the performance of predictive distribution models,” *Global Ecol. Biogeogr.*, vol. 17, no. 2, pp. 145–151, Mar. 2008.
- [40] S. Han, K. Zhu, M. Zhou, H. Alhumade, and A. Abusorrah, “Locating multiple equivalent feature subsets in feature selection for imbalanced classification,” *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 9, pp. 9195–9209, Sept. 2023.

- [41] Y. Xie, G. Liu, C. Yan, C. Jiang, M. Zhou, and M. Li, "Learning transactional behavioral representations for credit card fraud detection," *IEEE Trans. Neural Netw. Learn. Syst.*, 2022. DOI: 10.1109/TNNLS.2022.3208967
- [42] Y. Xie, G. Liu, C. Yan, C. Jiang, and M. Zhou, "Time-aware attention-based gated network for credit card fraud detection by extracting transactional behaviors," *IEEE Trans. Comput. Soc. Syst.*, vol. 10, no. 3, pp. 1004–1016, Jun. 2023.
- [43] S. Han, K. Zhu, M. Zhou, and X. Cai, "Competition-driven multimodal multiobjective optimization and its application to feature selection for credit card fraud detection," *IEEE Trans. Syst. Man Cybern. Syst.*, vol. 52, no. 12, pp. 7845–7857, Dec. 2022.
- [44] S. Han, K. Zhu, M. Zhou, and X. Liu, "Joint deployment optimization and flight trajectory planning for UAV assisted IoT data collection: A bilevel optimization approach," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 11, pp. 21492–21504, Nov. 2022.
- [45] Z. Huang, Y. Liu, C. Zhan, C. Lin, W. Cai, and Y. Chen, "A novel group recommendation model with two-stage deep learning," *IEEE Trans. Syst. Man Cybern. Syst.*, vol. 52, no. 9, pp. 5853–5864, Sept. 2022.
- [46] M. Cui, L. Li, M. Zhou, and A. Abusorrah, "Surrogate-assisted autoencoder-embedded evolutionary optimization algorithm to solve high-dimensional expensive problems," *IEEE Trans. Evol. Comput.*, vol. 26, no. 4, pp. 676–689, Aug. 2022.
- [47] M. Cui, L. Li, M. Zhou, J. Li, A. Abusorrah, and K. Sedraoui, "A Bi-population cooperative optimization algorithm assisted by an autoencoder for medium-scale expensive problems," *IEEE/CAA J. Autom. Sinica*, vol. 9, no. 11, pp. 1952–1966, Nov. 2022.
- [48] Z. Zhao, S. Liu, M. Zhou, and A. Abusorrah, "Dual-objective mixed integer linear program and memetic algorithm for an industrial group scheduling problem," *IEEE/CAA J. Autom. Sinica*, vol. 8, no. 6, pp. 1199–1209, Jun. 2021.
- [49] X. Zhu and M. Zhou, "Multiobjective optimized deployment of edge-enabled wireless visual sensor networks for target coverage," *IEEE Internet Things J.*, vol. 10, no. 17, pp. 15325–15337, Sept. 2023.
- [50] D. Li, Q. Wu, M. Zhou, and F. Luo, "HHFS: A hybrid hierarchical feature selection method for ageing gene classification," *IEEE Trans. Cognit. Dev. Syst.*, vol. 15, no. 2, pp. 690–699, Jun. 2023.
- [51] H. Liu, M. Zhou, X. Lu, A. Abusorrah, and Y. Al-Turki, "Analysis of evolutionary social media activities: Pre-vaccine and post-vaccine emergency use," *IEEE/CAA J. Autom. Sinica*, vol. 10, no. 4, pp. 1090–1092, Apr. 2023.
- [52] W. Duo, M. Zhou, and A. Abusorrah, "A survey of cyber attacks on cyber physical systems: Recent advances and challenges," *IEEE/CAA J. Autom. Sinica*, vol. 9, no. 5, pp. 784–800, May 2022.
- [53] X. Luo, Y. Yuan, S. Chen, N. Zeng, and Z. Wang, "Position-transitional particle swarm optimization-incorporated latent factor analysis," *IEEE Trans. Knowl. Data Eng.*, vol. 34, no. 8, pp. 3958–3970, Aug. 2022.
- [54] D. Wu, Y. He, X. Luo, and M. Zhou, "A latent factor analysis-based approach to online sparse streaming feature selection," *IEEE Trans. Syst. Man Cybern. Syst.*, vol. 52, no. 11, pp. 6744–6758, Nov. 2022.
- [55] X. Luo, H. Wu, and Z. Li, "Neulft: A novel approach to nonlinear canonical polyadic decomposition on high-dimensional incomplete tensors," *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 6, pp. 6148–6166, Jun. 2023.



Honghao Zhu (Member, IEEE) received the B.S. degree in computer science from Huaibei Normal University in 2003, M.S. degree from University of Electronic Science and Technology of China in 2009, and the Ph.D. degree in computer science and technology from Tongji University in 2021. He is an Associate Professor at Bengbu University. His research interests include machine learning, evolutionary algorithms, recommendation systems, and credit card fraud detection.



MengChu Zhou (Fellow, IEEE) received the B.S. degree in control engineering from Nanjing University of Science and Technology, in 1983, M.S. degree in automatic control from Beijing Institute of Technology, in 1986, and Ph.D. degree in computer and systems engineering from Rensselaer Polytechnic Institute, USA in 1990. He joined the Department of Electrical and Computer Engineering, New Jersey Institute of Technology in 1990, and has been a Distinguished Professor since 2013. His research interests include intelligent automation, robotics, Petri nets, Internet of Things, edge/cloud computing, and big data analytics. He has over 1200 publications including 17 books, over 800 journal papers including over 650 IEEE Transactions papers, 31 patents and 32 book-chapters. He is a recipient of Excellence in Research Prize and Medal from NJIT, Humboldt Research Award for US Senior Scientists from Alexander von Humboldt Foundation, and Franklin V. Taylor Memorial Award and the Norbert Wiener Award from IEEE Systems, Man, and Cybernetics Society, and Edison Patent Award from the Research & Development Council of New Jersey. He is a Life Member of Chinese Association for Science and Technology-USA and served as its President in 1999. He is Fellow of IEEE, (International Federation of Automatic Control (IFAC), American Association for the Advancement of Science (AAAS), Chinese Association of Automation (CAA) and National Academy of Inventors (NAI)).



Yu Xie received the B.S. degree in information security from Qingdao University in 2017, the Ph.D. degree in computer software and theory from Tongji University in 2022. He joined the College of Information Engineering, Shanghai Maritime University, in 2022. His research interests include credit card fraud detection, machine learning, representation learning, and big data.



Aiiad Albeshri Received the M.S. and the Ph.D. degrees in information technology from Queensland University of Technology, Australia in 2007 and 2013, respectively. He has been an Associate Professor at the Department of Computer Science, King Abdulaziz University, Saudi Arabia since 2018. His research interests include information security, trust in cloud computing, big data and high-performance computing (HPC).