Hacettepe Journal of Mathematics & Statistics

Hacet. J. Math. Stat.
Volume 53 (3) (2024), 879 – 896
DOI : 10.15672/hujms.1346686

Research Article

Particle swarm optimization based feature selection using factorial design

Emre Koçak*[1,2], Hacı Hasan Örkcü[1]

[1] *Department of Statistics, Faculty of Sciences, Gazi University, Ankara, Turkey*
[2] *Graduate School of Natural and Applied Sciences, Gazi University, Ankara, Turkey*

Abstract

Feature selection, a common and crucial problem in current scientific research, is a crucial data preprocessing technique and a combinatorial optimization task. Feature selection aims to select a subset of informative and appropriate features from the original feature dataset. Therefore, improving performance on the classification task requires processing the original data using a feature selection strategy before the learning process. Particle swarm optimization, one of the metaheuristic algorithms that prevents the growth of computing complexity, can solve the feature selection problem satisfactorily and quickly with appropriate classification accuracy since it has local optimum escape strategies. There are arbitrary trial and error approaches described separately in the literature to determine the critical binary particle swarm optimization parameters, which are the inertial weight, the transfer function, the threshold value, and the swarm size, that directly affect the performance of the binary particle swarm optimization algorithm parameters used in feature selection. Unlike these approaches, this paper enables us to obtain scientific findings by evaluating all binary particle swarm optimization parameters together with the help of a statistically based factorial design approach. The results show how well the threshold and the transfer function have statistically affected the binary particle swarm optimization algorithm performance.

Mathematics Subject Classification (2020). 62K15, 62H30, 65K05, 68T20, 90C59

Keywords. Feature selection, classification, particle swarm optimization, chaotic map, transfer function, factorial design

1. Introduction

Prediction using train data and useful features is one of the fundamental goals of data modeling and classification. Large datasets with high dimensionality and a comparatively small number of samples constitute a severe problem for machine learning applications. Recent advances in science and technology have resulted in a rapid increase in the amount of data. As a result, these application methods typically deal with cases with thousands of features. In the literature, the term used for this problem is the curse of dimensionality [5,6]. Reducing the dimensionality of datasets is crucial for machine learning methods such

*Corresponding Author.
Email addresses: emrekocak@gazi.edu.tr (E. Koçak), hhorkcu@gazi.edu.tr (H.H. Örkcü)
Received: 20.08.2023; Accepted: 01.05.2024

as classification [7, 43]. Under the topic of feature extraction and feature selection, two main techniques for dimensionality reduction have been recommended to improve classification accuracy (ACC) performance and reduce computational complexity [54,62]. In the feature extraction process, fewer features are produced by combining existing features, and as a result, these features include the majority of the data present in the original features. However, a subset of the original features is selected by removing meaningless and unnecessary features in feature selection.

The best feature sets are usually chosen using filter, wrapper, embedding, and hybrid approaches. The filter approach assesses feature importance without utilizing any learning method. As a result, this approach produces models that are frequently quick. According to information criteria, the features are assessed and ordered in this approach, and the features with the highest value are selected [30]. There are two types of filter-based approaches: univariate and multivariate. According to a particular criterion, information gain [42], mutual information [64], Laplacian score [21], and Fisher score [19], the univariate filter approach considers the suitability of features to the target class. On the other hand, a multivariate filter approach has been proposed that addresses both unrelated and unnecessary features in sorting techniques and aims to evaluate feature dependencies to some extent, such as minimal redundancy maximal relevance [38], relevance redundancy feature selection [17], normalized mutual information feature selection [15], and Hilbert-Schmidt independence criterion [70].

In search processes, a specific learning model is utilized to assess a subset of features to select a feature set with the highest ACC in the wrapper approach, where a classifier such as the Support Vector Machine (SVM) is utilized and trained to assess several significant features. Most wrapper approaches use iterative search processes, in which each iteration of the learning model is used to direct the population of solutions in the direction of the best solution. However, these models often incur high computational costs since the wrapper approach is a learning model in the search process [45].

The hybrid approach is a compound of models that tries to take advantage of the filter and wrapper approaches. This approach primarily concentrates on attaining the best possible performance with a particular learning algorithm and time complexity, like a filter-based approach. The embedded approach considers the feature selection issue as a part of the machine learning method. In this approach, the final feature subset is searched using a machine learning method.

The selection of a feature subset is an NP-hard problem. The total search space to find the most related and non-redundant features, including all possible subsets, is 2^n, where n is the number of original features. The most suitable features are determined through search; however, this is typically not computationally possible. Searching for a solution is both computationally feasible and high-quality, as evaluating every possible subset would be highly costly.

The simplest method is finding the best subset by evaluating each feasible subset using a thorough search approach. This approach ensures an optimal feature subset, but it takes much work to identify the optimal. The metaheuristic approaches, which include different nature-inspired algorithms such as Particle Swarm Optimization (PSO), are methods for solving NP-hard and complex optimization problems. These algorithms are frequently used in feature selection problems to prevent the growth of computing complexity [3,52,63]. In addition, these algorithms can satisfactorily and quickly resolve the issue of feature selection with the convenient ACC since these algorithms have strategies to escape from the local optimum [45].

We acknowledged a wrapper approach by using PSO for the feature selection problems. The PSO algorithm discussed in this paper is a powerful optimization method. The selection of the final feature subset has recently been optimized by using this optimization method by several researchers. Huang and Dun [23] proposed a hybrid approach for parameter optimization and performance improvement of feature selection using a combination of PSO and SVM. Unler et al. [58] presented a novel search method that merged the filter approach with the PSO-based wrapper approach to select the final feature subset. Xue et al. [67] introduced unique starting approaches and the best particle update techniques to reduce computing complexity. Moradi and Gholampour [36] aimed to develop a new local search method that includes the PSO algorithm to select a subset of related and non-redundant features by integrating filter and wrapper approaches. Jain et al. [25] eliminated unnecessary and unrelated features by combining the correlation feature selection method with a modified binary PSO algorithm to choose a highly relevant feature subset. Qasim and Algoma [40] suggested a novel PSO-based feature selection approach combining the regression model with Bayesian information criteria. Prasad et al. [39] combined the advanced wrapper-based approach recursive PSO algorithm with the various feature selection strategies based on a filter-based approach. A PSO-based feature selection approach using several classifiers was suggested by [69] to increase ACC while lowering computing costs. A novel graph-based feature selection technique to increase ACC has been created by [44]. This approach suggests a novel methodology for initializing particles in the PSO algorithm based on the node centrality requirement. A new particle ordering based on particle distance from dominant and non-dominant PSO particles was proposed and then used for feature order calculation by [50].

The selection of algorithm parameters determines the computational performance of the metaheuristic algorithms. Like most metaheuristic techniques, the PSO algorithm is applied in various fields, but unlike other algorithms, it does not have precise guidelines for determining algorithm parameters. The algorithm's performance can be significantly impacted by varying the parameter settings in PSO. In this regard, it is essential to ascertain the appropriate values for these parameters [14]. In studies on PSO, its parameters are usually defined using a trial-and-error approach or specified intuitively. To the best of our knowledge, there needs to be a study where the statistical effects of the parameters that influence the choice of the PSO algorithm for the final feature subset are evaluated together, although there are separately well-studied PSO studies for arbitrary problems in the literature.

This paper aims to discover a statistical answer to the problem mentioned above through the factorial design approach. The factorial design, one of the most popular and frequently applied statistical techniques in experimental studies, determines the most crucial PSO parameters for any optimization problem. In the selection of the final feature subset, the use of statistical techniques provides a significant contribution to determining the key PSO parameters, which are the inertial weight, the transfer function, the threshold value, and the swarm size. For this purpose, factorial analysis of variance (ANOVA) is used to identify important PSO parameters that have a high efficiency on PSO performance. In addition, the performance of the metaheuristic can be improved since this method enables statistical interpretation of the parameters' importance. In this article, where an experiment with 4^4 factors is discussed, the factors, each with four levels, are the key PSO parameters mentioned above.

The balance between the ability to search locally and globally is controlled by the inertia weight, as proposed by [51]. The global search is made easier with a large inertia weight, and the local search is made easier with a small one. It is crucial to adjust the inertia weight value correctly. The primary determinant of convergence, the inertia weight, will significantly impact the PSO search procedure. The above-mentioned early convergence of the PSO process is frequently caused by particle entrapment in a local optimum [11,65].

Kennedy and Eberhart first developed the idea of a transfer function, which enables the PSO algorithm to operate in the binary searching space [28]. Transfer functions are required to map a continuous search space to a discrete space. Because the transfer function does not affect the algorithm's computational cost, it is independent of the algorithm, and can facilitate its discovery and use, and determining it is a crucial step in optimizing the BPSO algorithm's performance [4, 24, 33]. For these reasons, various functions have been proposed in the literature to examine the effects of these functions on the BPSO algorithm. Additionally, the advantages and disadvantages of these proposed functions against each other have been examined.

Since the transfer function aims to express the probability that a position vector element will go from 0 to 1, it must be limited to the interval $[0, 1]$. The Sigmoid function given in Equation (2.3) is utilized in the original BPSO algorithm to convert the continuous variable into binary. As a result, the sigmoid function's ability to work is crucial to the BPSO's performance. Values in the interval $[0, 1]$ obtained with the sigmoid function take 0 or 1 according to a particular threshold value. If the value in the interval $[0, 1]$ is less than the threshold value, it is converted to 0; if it is greater than the threshold value, it is converted to 1. In the original BPSO, this threshold value is a random number distributed equally over $[0, 1]$ and is expressed with *rand()*, as seen in Equation (2.3).

Swarm size is the number of members in the population. The swarm size indicates the number of solutions analyzed at each iteration. It is common knowledge that the number of particles should be chosen based on the specifics of the problem [26]. Higher swarm size numbers can provide higher-quality outcomes. However, there is a trade-off between swarm size, computational time, and solution quality [2]. According to [59], the computation time of the algorithm can occur for large swarm sizes. However, the PSO algorithm might conclude with a local minimum if few particles are used. But at the same time, empirical evidence has demonstrated that the PSO can identify the best solutions at even small swarm sizes [8].

In Section 2, the PSO, the BPSO, and its parameters will be discussed in detail. Section 3 is devoted to the 4^4 factorial design. Section 4 presents the experimental results. The conclusion is reported in Section 5.

2. Methods

The performance of the PSO algorithm used in selecting the final feature subset is affected by many factors. In addition to the BPSO algorithm, the chaotic maps used instead of the inertial weight in previous studies to improve the performance of this algorithm and transfer functions that significantly affect feature selection will be discussed in detail below.

2.1. Particle swarm optimization (PSO)

The PSO algorithm, introduced by [27], is a population-based stochastic algorithm and mighty swarm intelligence-based optimization method. The fundamental concept of the PSO is that knowledge is best optimized by social interaction among the population as well as personal experience [66].

In a d-dimensional search space, the position and velocity of particle i are represented as the vectors $X_i = (x_{i1}, x_{i2}, ..., x_{id})$ demonstrate particles position and $V_i = (v_{i1}, v_{i2}, ..., v_{id})$ demonstrate the particles flight velocity over a solution space in the PSO algorithm. Each particle in the swarm is given a score using a function to determine its fitness value, which indicates how well it solves the problem. Each particle keeps track of its own *pbest* and *gbest*. Then, updated rules for new positions are applied to all particles in the d-dimensional search space until the global optimal position is discovered. The modified velocity and position of each particle can be calculated in Equation (2.1) and Equation (2.2), respectively [13]:

$$V_i^{t+1} = \omega V_i^t + c_1 R_{rand_1}(pbest_i^t - X_i^t) + c_2 R_{rand_2}(gbest^t - X_i^t) \tag{2.1}$$

$$X_i^{t+1} = X_i^t + V_i^{t+1} \tag{2.2}$$

where V_i^t velocity of particle i at iteration t, ω inertia weight, c_1, c_2 acceleration coefficients, R_{rand_1}, R_{rand_2} are random numbers uniformly distributed $(0, 1)$, X_i^t position of particle i at iteration t, $pbest_i^t$ best position of particle i until iteration t, $gbest^t$ best position of the group until iteration t.

A predetermined maximum velocity (V_{max}) and the new velocity $V_i^t \in [-V_{max}, V_{max}]$ both provide restrictions on the velocity. In this paper, $V_{max} = 6$ is considered. Therefore, V_i^t is limited to $[-6, 6]$ values. The algorithm is completed when a predefined good fitness value or a maximum number of iterations is reached [66].

2.2. Binary PSO

Initially, the PSO algorithm was used to solve problems in the continuous search space. Like many other optimization issues, feature selection occurs in discrete search space. Kennedy and Eberhart [28] improved the binary PSO (BPSO) algorithm to address optimization issues in discrete domains. The velocity is still updated in BPSO as it is in the conventional PSO algorithm. However, the values of X_i, $pbest_i$, and $gbest$ can only be 0 or 1. Hence the velocity indicates the probability of a particle in the position vector getting the value 1 [68].

Based on the probability value $S(V_i^t)$ acquired from Equation (2.4), the position of the current particle is updated in BPSO as in Equation (2.3).

$$X_i^{t+1} = \begin{cases} 1, & \mathit{rand()} \leq S\left(V_i^{t+1}\right) \\ 0, & \text{otherwise} \end{cases} \tag{2.3}$$

$$S\left(V_i^{t+1}\right) = \frac{1}{1 + e^{-V_i^{t+1}}} \tag{2.4}$$

where $S(V_i^t)$ is the Sigmoid function and *rand()* is random numbers uniformly distributed (0,1).

2.3. Chaotic maps

The search process of the PSO algorithm will be significantly impacted by the inertia weight ω, which is the main factor driving convergence. Throughout the algorithm process, the inertia weight is dynamically changed based on feedback regarding the optimal placements of the particles for exploration and exploitation. This dynamic adjustment of the search capability is achieved by dynamically modifying the inertia weight [37]. There are certain proposed PSO models that cannot effectively use the conventional inertia weight adaption process. The search process of the PSO algorithm frequently experiences particle entrapment in a local optimum, which results in the above-mentioned premature convergence [12]. A large inertia weight helps with the global search, whereas a small one helps with the local search. Therefore, the inertia weight value must be defined correctly for the algorithm to perform appropriately. Various processes have been proposed over time, including chaotic maps, time-varying inertia weights, and constant and random inertia weights. In this paper, chaotic maps are utilized to avoid early convergence.

Chaos, generally a dynamic, nonconvergent, and deterministic method, can be described as a fact in which any small change in the initial condition can give rise to a nonlinear change in future behavior. Additionally, it is described as a semi-random behavior generated by nonlinear deterministic systems [55]. Although chaos appears random and unpredictable, it also has a particular element of regularity [1]. The chaotic variables must have these properties to cycle through all possible states within a specific range without repeating any of them. Consequently, instead of traditional stochastic search, chaos search can avoid a local optimal solution and increase the convergence rate.

Four different chaotic maps commonly utilized in numerous studies on feature selection to create chaotic clusters are discussed in this paper [11, 16, 22, 41, 47, 53, 57]. The four applied chaotic maps are listed in Table 1, along with their mathematical definitions.

Table 1 demonstrates these maps, where w_t denotes the *t-th* number in the chaotic set and t is defined as the index of the chaotic set w. The other parameters including c, d, and μ are the control parameters. The chaotic behavior of the dynamic system is assessed using these parameters. According to [46, 48, 49], we adjust the beginning value w_0 to 0.7 for all chaotic maps. Four chaotic maps that were applied and visualized across 100 iterations are shown in Figure 1.

Table 1. Types of chaotic map

Chaotic map	Mathematical formula	Value range
Circle	$w_{t+1} = mod\left(w_t + d - \left(\frac{c}{2\pi}\right) sin\left(2\pi w_t\right), 1\right), c = 0.5 \text{ and } d = 0.2$	(0,1)
Logistic	$w_{t+1} = c\, w_t \left(1 - w_t\right), c = 4$	(0,1)
Piecewise	$w_{t+1} = \begin{cases} \frac{w_t}{l}, & \text{if } 0 \leq w_t < l \\ \frac{w_t - l}{0.5 - l}, & \text{if } l \leq w_t < 0.5 \\ \frac{1 - l - w_t}{0.5 - l}, & \text{if } 0.5 \leq w_t < 1 - l \\ \frac{1 - w_t}{l}, & \text{if } 1 - l \leq w_t < 1 \end{cases}, l = 0.4$	(0,1)
Singer	$w_{t+1} = \mu\left(7.86 w_t - 23.31 w_t^2 + 28.75 w_t^3 - 13.302875 w_t^4\right), \mu = 1.07$	(0,1)

2.4. Transfer function

A transfer function must map a continuous search space to a discrete space. Many studies suggest using transfer functions since they are independent of the algorithm, have no impact on the algorithm's computational cost, and promote algorithm exploration and exploitation [4, 24, 33]. In this regard, choosing an appropriate transfer function is an essential decision to improve the performance of the BPSO algorithm.

In the early stages of the selection process, focusing on exploration is generally more significant than thoroughly intensifying the examination of promising feature space regions. However, exploitation becomes increasingly crucial in the later phases since we need to raise the probability of finding better solutions that are similar to those found in the earlier phases [32]. The continuous search space is converted into the discrete binary space based on the properties of the BPSO algorithm, as mentioned in Section 2.2. The transfer function must be restricted to a range of $[0, 1]$ since the purpose is to represent the probability that the element of the position vector goes from 0 to 1. It has been claimed that the original BPSO algorithm fails to provide a good balance between exploration and exploitation due to the limitations of the Sigmoid transfer function [24, 31, 56, 61]. To overcome this, modified BPSO with different transfer functions has been used in the literature [10, 20, 24, 29, 34].

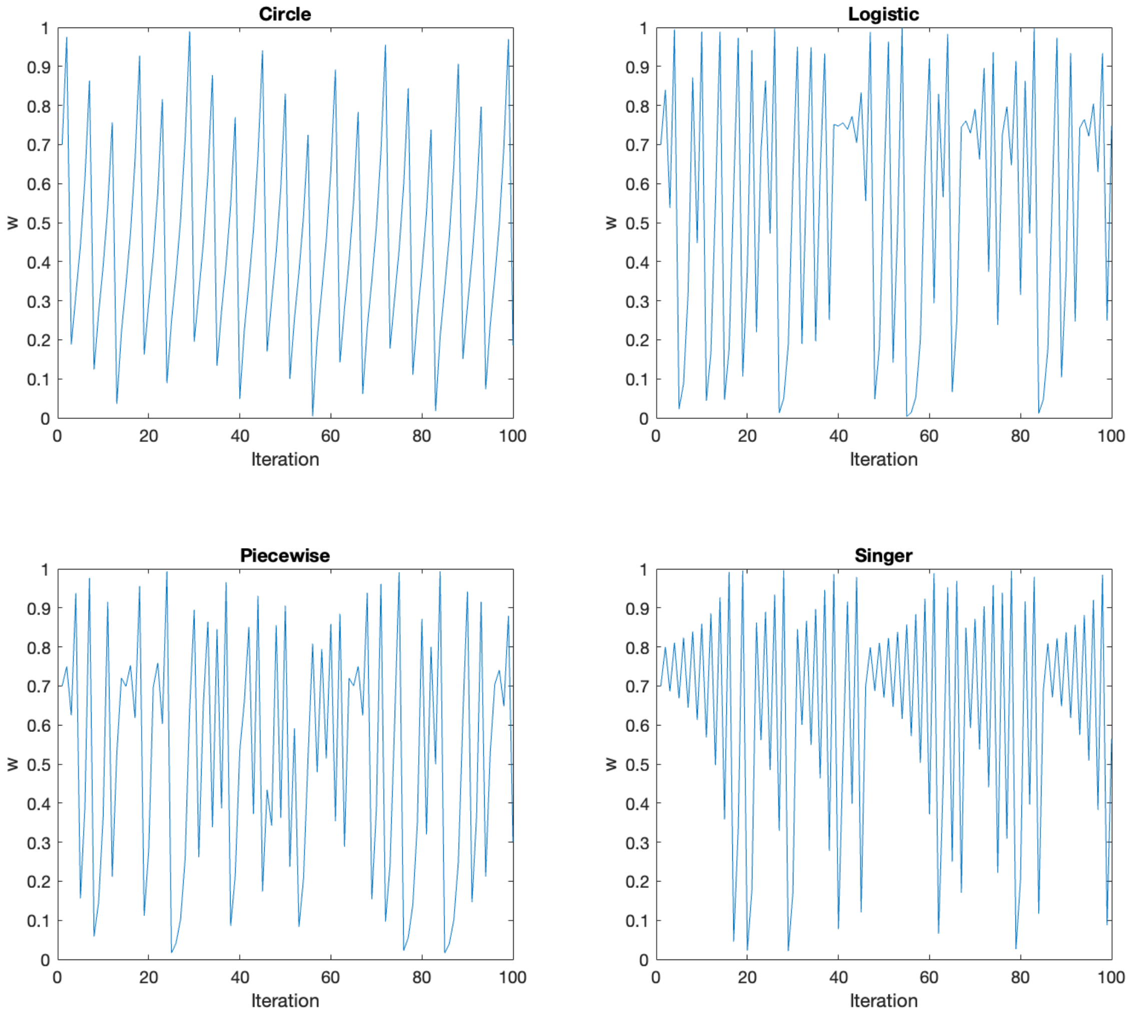

Figure 1. Visualization of chaotic maps

The mathematical definitions of the four different transfer functions discussed in this paper is given in Table 2 and the visualization of these functions are given in Figure 2.

Table 2. Types of transfer function

Transfer function	Mathematical formula
S-shaped	$S\left(V_i^{t+1}\right) = \frac{1}{1+e^{-V_i^{t+1}}}$
V-shaped	$V\left(V_i^{t+1}\right) = \left\|tanhV_i^{t+1}\right\|$
U-shaped	$U\left(V_i^{t+1}\right) = \alpha\left\|\left(V_i^{t+1}\right)^{\beta}\right\|,\ \alpha = 1, \beta = 2$
Z-shaped	$Z\left(V_i^{t+1}\right) = \sqrt{\left(1 - \alpha^{V_i^{t+1}}\right)},\ \alpha = 2$

2.5. Threshold

Since the transfer function aims to express the probability that a position vector element will go from 0 to 1, it must be limited to the interval $[0, 1]$. The Sigmoid function given

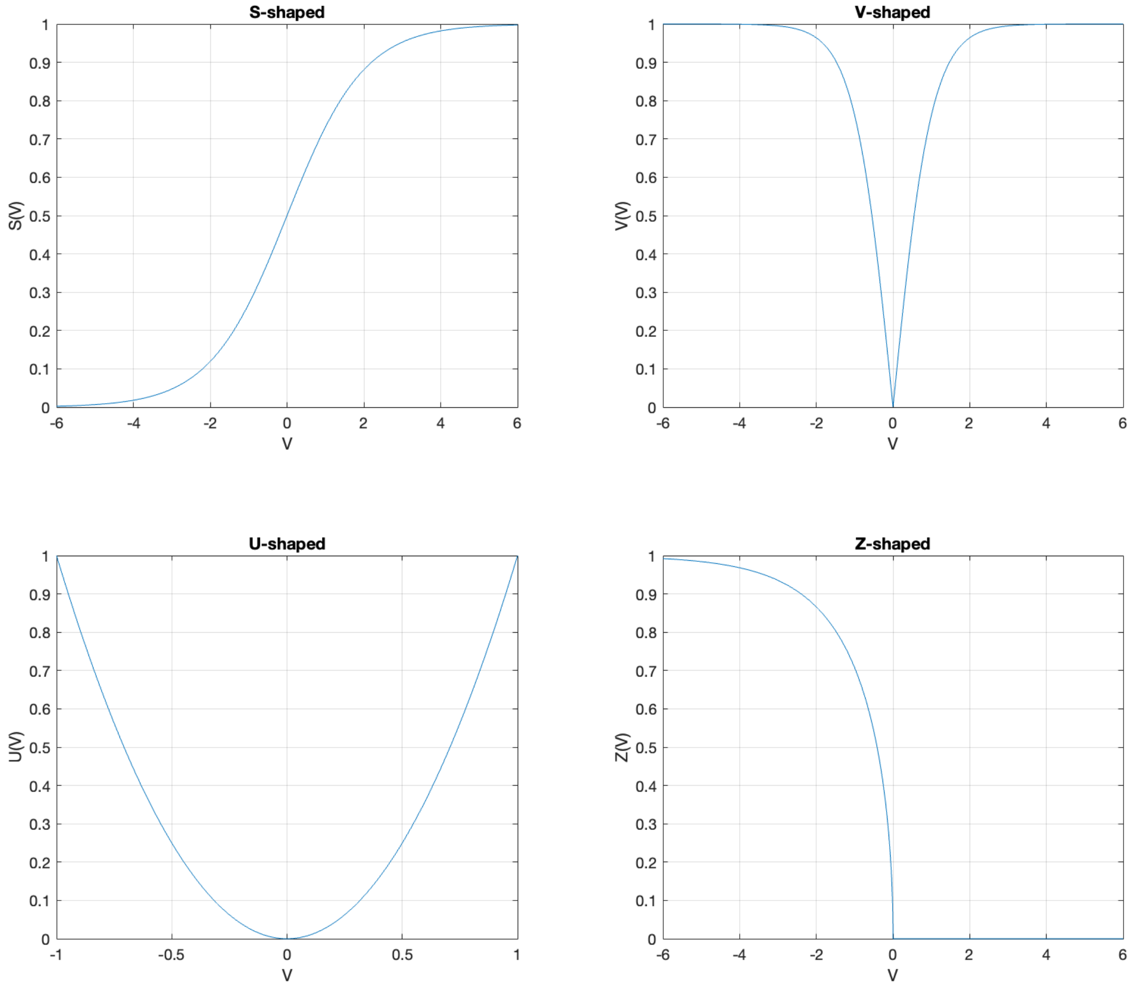

Figure 2. Visualization of transfer functions

in Equation (2.3) is utilized in the original BPSO algorithm to convert the continuous variable into binary. As a result, the sigmoid function's ability to work is crucial to the BPSO's performance. Values in the interval $[0, 1]$ obtained with the sigmoid function take 0 or 1 according to a particular threshold value. If the value in the interval $[0, 1]$ is less than the threshold value, it is converted to 0; if it is greater than the threshold value, it is converted to 1. In the original BPSO, this threshold value is a pseudorandom number distributed equally over $[0, 1]$ and is expressed with *rand()*, as seen in Equation (2.3). However, this random value often leads to selecting unfit features and rejecting potential features. In this case, the convergence of the BPSO algorithm will be delayed, and the chances of getting an optimal solution will decrease.

Let us explain this situation with an example. Although $S\left(V_i^{t+1}\right)$ is a feature that has a high value of 0.9, it can be discarded for the next iteration if the generated random value is greater than 0.9. Although $S\left(V_i^{t+1}\right)$ is a feature with a value as low as 0.05, it can also be selected for the next iteration if the generated random value is less than 0.05. Secondly, if generated random value is very small, like 0.001, then features with values of $S\left(V_i^{t+1}\right)$ as 0.005, 0.4, or 0.99 will have an equal chance to be selected for the next iteration.

3. The 4^4 factorial design

Factorial design was first used by [18] and [71] to simultaneously examine the effects of numerous factors. It is more efficient and less expensive than dealing with each factor separately. This feature will bring to the fore factorial designs in choosing the best factors influencing feature selection. In this paper, we consider an experiment with 4^4 factors, as four factors have four levels each due to their widespread use in the literature. The factors and their levels are given in Table 3.

Table 3. The factors and their levels

Factors	Levels			
A: Chaotic map	Circle	Logistic	Piecewise	Singer
B: Transfer function	S-shaped	V-shaped	U-shaped	Z-shaped
C: Threshold	0.5	0.6	0.7	0.8
D: Swarm size	10	25	50	100

3.1. Statistical model

Equation (3.1) provides the statistical model for the 4^4 factorial experiment.

$$\begin{aligned} y_{ijklm} = {} & \mu + \alpha_i + \beta_j + \gamma_k + \delta_l + \alpha\beta_{ij} + \alpha\gamma_{ik} + \alpha\delta_{il} + \beta\gamma_{jk} + \beta\delta_{jl} + \gamma\delta_{kl} \\ & + \alpha\beta\gamma_{ijk} + \alpha\beta\delta_{ijl} + \alpha\gamma\delta_{ikl} + \beta\gamma\delta_{jkl} + \alpha\beta\gamma\delta_{ijkl} + \varepsilon_{ijklm}, \\ & i, j, k, l = 1, 2, 3, 4, \quad m = 1, 2, ..., r \end{aligned} \tag{3.1}$$

Here, y_{ijklm} is the response variable, μ is the overall mean, α_i is the effect of *i-th* level of the inertia weight factor, β_j is the effect of *j-th* level of the transfer function factor, γ_k is the effect of *k-th* level of the threshold factor, δ_l is the effect of *l-th* level of the swarm size factor, and r is the replication number. In addition, $\alpha\beta_{ij}$, $\alpha\gamma_{ik}$, $\alpha\delta_{il}$, $\beta\gamma_{jk}$, $\beta\delta_{jl}$, and $\gamma\delta_{kl}$ are the 2-factors interaction effects, $\alpha\beta\gamma_{ijk}$, $\alpha\beta\delta_{ijl}$, $\alpha\gamma\delta_{ikl}$, and $\beta\gamma\delta_{jkl}$ are the 3-factors interaction effects, $\alpha\beta\gamma\delta_{ijkl}$ is the 4-factors interaction effect among the related factors, and ε_{ijklm} is the error term, see [35] for further details.

3.2. Hypotheses

Testing the main effects and their interactions is the goal of factorial design. The general form of the null hypothesis can thus be expressed as follows [2]:

H_0 : The main effect of the factor (or interaction effect of the factors of interest) is statistically insignificant.

H_0 indicates no significant difference among the levels of a factor (or there is no significant interaction between the factors of interest). It is also evident that fifteen null hypotheses exist for the 4^4 factorial experiments implemented in this article. These hypotheses are respectively: four main effects (A, B, C, and D), six 2-factor interactions (AB, AC, AD, BC, BD, and CD), four 3-factor interactions (ABC, ABD, ACD, and BCD), and one 4-factor interactions (ABCD).

3.3. Test statistics and decision rule

We used the presented factorial design to investigate the main effects and interactions in this paper. In factorial ANOVA, the error terms are assumed to be independently and identically distributed normal with mean zero and variance σ^2. Under the normality assumption, F tests are used to test the fifteen null hypotheses in Table 4.

Table 4. Analysis of variance

Source	df	Sum of Square	Mean of Square	F-value
A	*a-1*	SS_A	$MS_A = \frac{SS_A}{a-1}$	$F_A = \frac{MS_A}{MS_{Error}}$
B	*b-1*	SS_B	$MS_B = \frac{SS_B}{b-1}$	$F_B = \frac{MS_B}{MS_{Error}}$
C	*c-1*	SS_C	$MS_C = \frac{SS_C}{c-1}$	$F_C = \frac{MS_C}{MS_{Error}}$
D	*d-1*	SS_D	$MS_D = \frac{SS_D}{d-1}$	$F_D = \frac{MS_D}{MS_{Error}}$
AB	*(a-1)(b-1)*	SS_{AB}	$MS_{AB} = \frac{SS_{AB}}{(a-1)(b-1)}$	$F_{AB} = \frac{MS_{AB}}{MS_{Error}}$
AC	*(a-1)(c-1)*	SS_{AC}	$MS_{AC} = \frac{SS_{AC}}{(a-1)(c-1)}$	$F_{AC} = \frac{MS_{AC}}{MS_{Error}}$
AD	*(a-1)(d-1)*	SS_{AD}	$MS_{AD} = \frac{SS_{AD}}{(a-1)(d-1)}$	$F_{AD} = \frac{MS_{AD}}{MS_{Error}}$
BC	*(b-1)(c-1)*	SS_{BC}	$MS_{BC} = \frac{SS_{BC}}{(b-1)(c-1)}$	$F_{BC} = \frac{MS_{BC}}{MS_{Error}}$
BD	*(b-1)(d-1)*	SS_{BD}	$MS_{BD} = \frac{SS_{BD}}{(b-1)(d-1)}$	$F_{BD} = \frac{MS_{BD}}{MS_{Error}}$
CD	*(c-1)(d-1)*	SS_{CD}	$MS_{CD} = \frac{SS_{CD}}{(c-1)(d-1)}$	$F_{CD} = \frac{MS_{CD}}{MS_{Error}}$
ABC	*(a-1)(b-1)(c-1)*	SS_{ABC}	$MS_{ABC} = \frac{SS_{ABC}}{(a-1)(b-1)(c-1)}$	$F_{ABC} = \frac{MS_{ABC}}{MS_{Error}}$
ABD	*(a-1)(b-1)(d-1)*	SS_{ABD}	$MS_{ABD} = \frac{SS_{ABD}}{(a-1)(b-1)(d-1)}$	$F_{ABD} = \frac{MS_{ABD}}{MS_{Error}}$
ACD	*(a-1)(c-1)(d-1)*	SS_{ACD}	$MS_{ACD} = \frac{SS_{ACD}}{(a-1)(c-1)(d-1)}$	$F_{ACD} = \frac{MS_{ACD}}{MS_{Error}}$
BCD	*(b-1)(c-1)(d-1)*	SS_{BCD}	$MS_{BCD} = \frac{SS_{BCD}}{(b-1)(c-1)(d-1)}$	$F_{BCD} = \frac{MS_{BCD}}{MS_{Error}}$
ABCD	*(a-1)(b-1)(c-1)(d-1)*	SS_{ABCD}	$MS_{ABCD} = \frac{SS_{ABCD}}{(a-1)(b-1)(c-1)(d-1)}$	$F_{ABCD} = \frac{MS_{ABCD}}{MS_{Error}}$
Error	*N-abcd*	SS_{Error}	$MS_{Error} = \frac{SS_{Error}}{N-abcd}$	
Total	*N-1*	SS_{Total}		

Here, df, SS, and MS denote degrees of freedom, sum of squares, and mean squares, defined as the ratio of the sum of squares to the degree of freedom, respectively. Additionally, each of the expressions a, b, c, and d indicates the number of levels of the relevant factors, and for 4^4 factorial experiments, it is $a = b = c = d = 4$. N is the total number of observations; for 4^4 factorial experiments, this number is 4^4r. Here, r is the number of repetitions. The F tests in the table have an F distribution with v_1 and v_2 degrees of freedom. Here v_1 represents the degrees of freedom for the factor (or interaction) of interest and $v_2 = N - abcd$ represents the degrees of freedom of the error. Note that for brevity, some formulas are not included here; see [35] for further details, such as SS or MS formulas.

The corresponding null hypothesis is rejected if the calculated value of F statistics is bigger than the corresponding critical value for the preassigned significance level α. Similarly, the importance of the effects can be assessed using the *p-value*. In other words, the null hypothesis is rejected if the *p-value* is smaller than the preassigned α [2].

4. Experimental results

This section presents an experiment with 4^4 factors results to determine the statistically significant PSO algorithm parameters used in feature selection and ACC. Considering that we presume each process will be repeated three times, the total number of the response variable is equivalent to $N = 4^4 x3 = 768$ ACC in the 4^4 factorial experiment. Thirty different runs are performed, and the average is taken for each ACC. Since the convergence behavior of the method is observed to be in a rapid convergence tendency, the number of iterations in each run is taken as 100. The ACC is calculated as in Equation (4.1):

$$ACC = \frac{TP + TN}{TP + TN + FP + FN} \tag{4.1}$$

where TP, TN, FP, and FN represent the number of true positives, true negatives, false positives, and false negatives, respectively.

In this paper, the Ionosphere dataset, frequently used in the literature, was used for experiments. This dataset consists of 34 features, two classes, and 351 patterns. In addition, some of these datasets contain features with a wide range of values. Thus, features with large values dominate features with small values. To overcome this situation, the datasets are normalized.

This paper presents a BPSO-based wrapper approach that uses the SVM classifier. The SVM, proposed by [60], is a supervised learning algorithm with the potential to tackle very large feature spaces. This is because the training of SVM is realized in such a way that the dimension of classified vectors does not have as much of an effect on the performance of SVM as it has on the performance of the conventional classifier. The SVM aims to seek a hyperplane as the decision surface to maximize the distance between a group of objects belonging to different classes.

The maximization of ACC and the minimization of the number of features are considered two opposing objectives in the multi-objective optimization problem of feature selection. The solution will be more successful if fewer features are selected and the ACC is higher. The particle's position determines the selection and rejection of features. If the position value is within (threshold,1], it indicates that the corresponding feature has been accepted; if not, it has been rejected [9]. Each position is evaluated using the previously determined fitness function, which calculates the ACC using the SVM and depends on the number of selected features. Equation (4.2) explains the fitness function described in this paper.

$$Fitness\ Function = min\left(\beta\ x\ (1 - ACC) + \gamma\ x\ \frac{SF}{NF}\right) \tag{4.2}$$

where SF and NF denote the number of selected features and the total number of features in the dataset, respectively. In addition, the significance of the ACC and the length of the selected feature subset are, respectively, represented by the two parameters β and γ, where $\beta, \gamma \in [0, 1]$ and $\gamma = 1 - \beta$. In this paper, $\beta = 0.9$ is taken. The ACC and the number of selected features of the factor levels considered are summarized in Table 5.

Table 6 lists the factors and interactions that are statistically significant and insignificant, together with their F statistics and corresponding *p-value*. According to the residual plots in Figure 3, the assumptions above-mentioned are satisfied, and the ANOVA results are reliable.

As seen in Table 6, it is evident that the main effects of B and C are statistically significant at $\alpha = 0.01$ level. In other words, the transfer function and the threshold are essential parameters that affect the performance of the BPSO algorithm used in feature selection and ACC. The threshold is the most significant parameter while the transfer function is the less significant in terms of *p-value* since the F value of the threshold has the highest value among the significant main effects. Additionally, the main effects of chaotic map (A) and swarm size (D) are not found to be statistically significant BPSO parameters.

Table 5. The results of the ACC and the number of selected features for factor levels

Chaotic maps	Transfer function	Swarm size	Threshold							
			0.5		0.6		0.7		0.8	
			ACC	SF	ACC	SF	ACC	SF	ACC	SF
Circle	S-shaped	10	0.9359	18.2556	0.9331	13.4667	0.9159	9.9889	0.9075	5.9889
		25	0.9349	17.8667	0.9277	13.5556	0.9174	9.6778	0.9185	6.2778
		50	0.9308	17.7667	0.9243	13.8667	0.9192	9.4778	0.9042	6.1667
		100	0.9356	18.4667	0.9303	14.0333	0.9149	9.5778	0.9334	5.9444
	U-shaped	10	0.9395	17.1222	0.9304	13.7778	0.9266	10.0889	0.9151	6.3889
		25	0.9439	16.6444	0.9285	13.5222	0.9265	10.1333	0.9153	6.9000
		50	0.9434	16.3667	0.9286	13.3000	0.9280	10.1556	0.9129	6.7556
		100	0.9307	16.6778	0.9323	13.8333	0.9329	10.0222	0.9156	6.8444
	V-shaped	10	0.9390	17.1000	0.9311	13.8889	0.9230	10.1667	0.9113	6.7222
		25	0.9318	16.6000	0.9329	13.9444	0.9253	10.3000	0.9166	6.7444
		50	0.9184	16.6889	0.9302	13.7333	0.9227	9.3667	0.9131	7.3000
		100	0.9471	17.6778	0.9300	13.6444	0.9254	10.2222	0.9212	6.8667
	Z-shaped	10	0.9340	17.4889	0.9229	12.9778	0.9183	10.0889	0.9151	6.7000
		25	0.9390	17.7333	0.9291	13.3889	0.9203	10.2667	0.9092	6.2889
		50	0.9421	17.9778	0.9245	13.7222	0.9161	9.5111	0.9063	6.5222
		100	0.9365	17.6333	0.9259	13.8444	0.9222	9.9889	0.9238	6.2222
Logistic	S-shaped	10	0.9351	18.1333	0.9304	13.4889	0.9171	9.7111	0.9342	6.2000
		25	0.9390	17.9444	0.9289	13.5778	0.9183	9.5889	0.9208	6.3778
		50	0.9355	18.5889	0.9279	14.1778	0.9171	10.1778	0.9059	6.1444
		100	0.9362	18.1111	0.9274	14.1778	0.9184	9.8333	0.9083	6.3556
	U-shaped	10	0.9391	17.2889	0.9283	13.1333	0.9282	10.0111	0.9171	6.6111
		25	0.9173	16.7444	0.9326	13.9778	0.9301	10.0444	0.9133	6.8556
		50	0.9455	16.9889	0.9267	13.5333	0.9252	10.3000	0.9146	6.5778
		100	0.9302	16.8222	0.9336	13.3556	0.9251	9.8889	0.9145	6.8889
	V-shaped	10	0.9306	17.1111	0.9338	13.5889	0.9181	9.8000	0.9147	7.4333
		25	0.9390	16.7667	0.9314	12.9889	0.9212	10.3333	0.9199	6.4778
		50	0.9335	16.7556	0.9332	13.2333	0.9225	10.0556	0.9174	7.0667
		100	0.9416	16.8556	0.9319	13.4111	0.9286	10.5889	0.9149	6.6556
	Z-shaped	10	0.9329	17.2889	0.9230	13.2333	0.9123	10.2889	0.9212	5.9222
		25	0.9370	17.6667	0.9234	13.4889	0.9177	10.1444	0.9338	6.3667
		50	0.9350	17.0556	0.9274	13.6000	0.9201	9.6778	0.9109	6.3222
		100	0.9380	17.8778	0.9222	13.6444	0.9203	10.0889	0.9053	6.4000
Piecewise	S-shaped	10	0.9351	18.2111	0.9329	13.6667	0.9171	9.7778	0.9337	6.4333
		25	0.9323	17.6778	0.9236	13.8667	0.9144	9.6222	0.9190	5.8222
		50	0.9339	18.2889	0.9313	14.0889	0.9153	9.2333	0.9220	6.0556
		100	0.9340	18.1778	0.9287	14.2667	0.9194	9.9667	0.9244	6.2333
	U-shaped	10	0.9166	17.2444	0.9355	13.4333	0.9255	10.2778	0.9144	7.0111
		25	0.9454	16.3556	0.9309	13.5222	0.9208	9.8556	0.9162	6.7667
		50	0.9295	17.2444	0.9308	13.5778	0.9260	10.1111	0.9157	6.7667
		100	0.9313	17.2000	0.9284	13.5556	0.9275	10.0000	0.9129	7.1222
	V-shaped	10	0.9392	18.0333	0.9350	13.3556	0.9240	10.7667	0.9220	6.6222
		25	0.9153	16.2000	0.9310	13.4444	0.9253	10.4111	0.9181	7.0333
		50	0.9339	16.5444	0.9329	13.9667	0.9248	10.1556	0.9187	7.3444
		100	0.9455	16.6778	0.9323	13.4444	0.9263	9.7778	0.9089	6.6000
	Z-shaped	10	0.9309	16.9000	0.9254	13.0111	0.9203	9.6556	0.9057	6.1333
		25	0.9368	17.3556	0.9258	13.2889	0.9196	9.3667	0.9014	6.3667
		50	0.9387	17.5222	0.9253	13.5444	0.9194	9.3778	0.9204	7.0111
		100	0.9400	18.1222	0.9253	13.8111	0.9199	10.0778	0.9091	6.2556
Singer	S-shaped	10	0.9289	18.1667	0.9267	13.8556	0.9159	10.1444	0.9087	6.3000
		25	0.9379	17.9111	0.9260	13.7889	0.9181	9.5222	0.9333	6.5000
		50	0.9340	18.3444	0.9309	14.0000	0.9151	10.1778	0.9048	6.0667
		100	0.9334	18.2889	0.9306	14.2333	0.9172	10.2333	0.9348	5.9000
	U-shaped	10	0.9213	16.1889	0.9287	13.6667	0.9267	10.2333	0.9094	6.7000
		25	0.9322	16.8889	0.9271	13.6667	0.9257	10.1778	0.9144	7.2333
		50	0.9287	16.8222	0.9325	13.2778	0.9226	9.8222	0.9205	6.7667
		100	0.9304	17.0111	0.9296	13.2111	0.9262	9.7778	0.9156	7.0111
	V-shaped	10	0.9283	17.2556	0.9335	13.0778	0.9222	9.8778	0.9135	6.7222
		25	0.9323	16.8222	0.9293	12.9556	0.9268	9.7000	0.9162	7.0889
		50	0.9336	16.9222	0.9278	13.4556	0.9248	10.0222	0.9138	6.5000
		100	0.9436	17.1778	0.9374	13.7111	0.9259	9.7889	0.9156	6.5333
	Z-shaped	10	0.9347	17.7111	0.9264	13.8556	0.9157	10.0222	0.9087	6.5000
		25	0.9349	18.1000	0.9249	13.7889	0.9147	9.7556	0.9172	6.0778
		50	0.9326	17.7889	0.9285	14.0000	0.9209	9.5778	0.9060	6.2444
		100	0.9378	17.8111	0.9244	13.7444	0.9200	10.2111	0.9080	6.1667

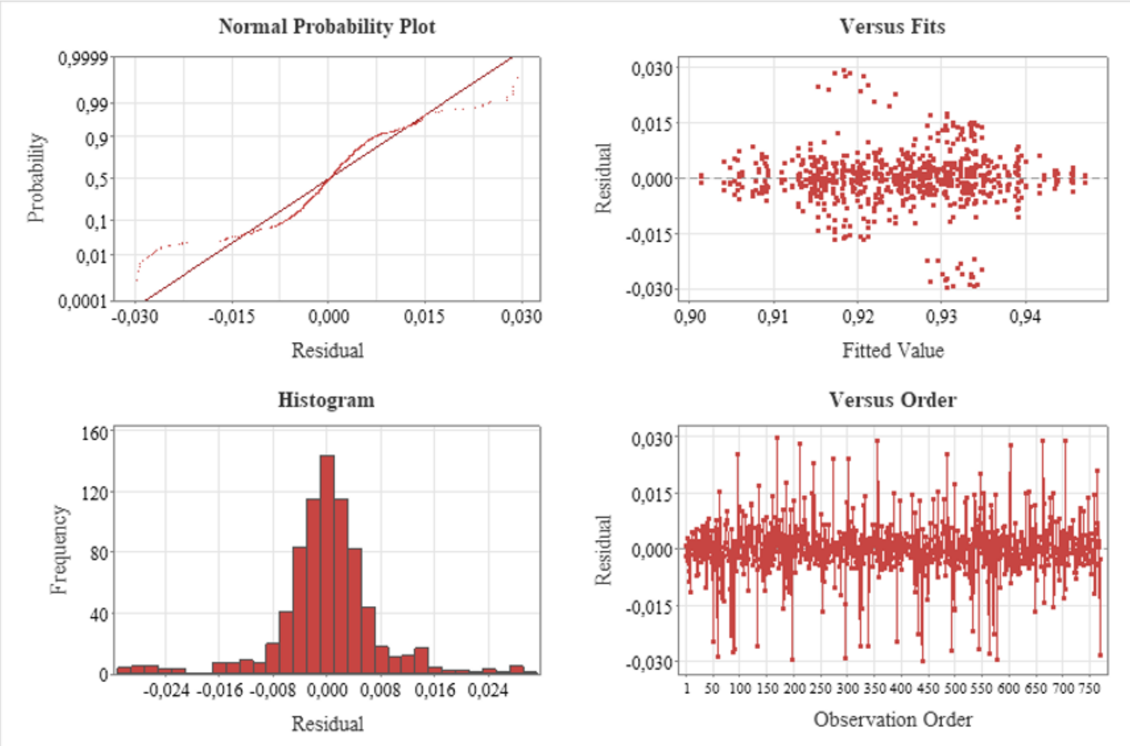

Figure 3. Residual plots for accuracy

Table 6. Results of 4^4 factorial experiment

Source	df	SS	MS	F-value	p-value
A	3	0.0001	0.0001	0.5621	0.6402
B	3	0.0014	0.0005	5.2427	0.0014
C	3	0.0394	0.0131	151.2853	0.0000
D	3	0.0005	0.0002	2.1022	0.0990
AB	9	0.0004	0.0001	0.5682	0.8234
AC	9	0.0004	0.0001	0.5179	0.8619
AD	9	0.0013	0.0001	1.7225	0.0811
BC	9	0.0043	0.0005	5.4969	0.0000
BD	9	0.0009	0.0001	1.1609	0.3180
CD	9	0.0008	0.0001	0.9924	0.4452
ABC	27	0.0014	0.0001	0.6149	0.9374
ABD	27	0.0020	0.0001	0.8358	0.7054
ACD	27	0.0029	0.0001	1.2212	0.2062
BCD	27	0.0027	0.0001	1.1620	0.2635
ABCD	81	0.0071	0.0001	1.0102	0.4597
Error	512	0.0444	0.0001		
Total	767	0.1101			

According to the 2-factor interaction comparison results at $\alpha = 0.01$ level, only BC interaction effects are statistically significant among all other interactions, as seen in Table

6. These results agree with the results obtained for the main effects. Namely, the transfer function and the threshold are the most significant BPSO parameters since 2-factors interactions include these parameters. Additionally, there is no statistically significant interaction at $\alpha = 0.01$ level among the 3-factor and 4-factor interactions.

Our research indicates that the threshold and the transfer function are the most significant parameters affecting the BPSO algorithm's performance in feature selection and ACC. The original BPSO algorithm considers the threshold value a random number. However, this paper shows the effect of different threshold values on feature selection and ACC. This situation can also be seen from the results in Table 5. The number of features decreases as the threshold value increases. In parallel with this, the accuracy rate decreases. The reason for this is the decrease in the number of features describing the model or the failure to select appropriate features that will increase the ACC by the BPSO algorithm. In addition, it is concluded that the transfer function is one of the most important reasons for not choosing appropriate features. The feature selection is impacted because different feature values result from using the inappropriate function. Therefore, focusing on the threshold and the transfer function, which are statistically significant in the 2-factor interaction, when defining BPSO parameters will significantly affect the algorithm's performance in feature selection and ACC. Additionally, this paper revealed that chaotic map and swarm size are not statistically efficient factors in the performance of the BPSO algorithm. Therefore, the chaotic map and the swarm size factor can be arbitrarily determined.

5. Conclusion

We recommend a novel approach based on the ANOVA technique with 4^4 factorial experiment, which are chaotic map, transfer function, threshold, and swarm size, to determine the critical BPSO parameters directly affecting the performance of the BPSO algorithm parameters used in feature selection and ACC. Unlike approaches based on arbitrarily defined intuitive selections or a trial-and-error approach, this statistically based approach enables us to derive scientific findings.

According to the findings, it is concluded that the threshold is the most critical parameter affecting the performance of the BPSO algorithm. The threshold value in the original BPSO method is regarded as a random number. However, the impact of various threshold values on feature selection and ACC was clearly demonstrated in this paper. As the threshold value increases, the number of features decreases, as seen in Table 5. This is accompanied by a decline in ACC. The decrease in the number of features describing the model or the failure to select the appropriate features that would improve the ACC using the BPSO algorithm are the causes of this. Additionally, it is seen that the transfer function is statistically significant. Therefore, focusing on the threshold and the transfer function when defining BPSO parameters will significantly affect the algorithm's performance in feature selection and ACC. The chaotic map and the swarm size, on the other hand, are not statistically significant.

Determining the parameters of feature selection metaheuristic methods is a very challenging problem. There are arbitrary trial-and-error approaches described separately in the literature to determine the critical BPSO parameters, which are the inertial weight, the transfer function, the threshold value, and the swarm size, that directly affect the performance of the BPSO algorithm parameters used in feature selection. Unlike these approaches, this paper allows us to obtain scientific findings by evaluating all BPSO parameters together with the help of a statistically based factorial design approach. The approach suggested in this paper can solve the feature selection problem of any other metaheuristic method. Considering statistical findings, the performance of the metaheuristic can thus be improved. Future studies can take this point into account.

Acknowledgements

The authors would like to thank Gazi University Academic Writing Application and Research Center for proofreading the article.

References

[1] B. Alatas, E. Akin and A.B. Ozer, *Chaos embedded particle swarm optimization algorithms*, Chaos, Solitons Fractals **40** (4), 1715-1734, 2009.
[2] B. Alptekin, S. Acitas, B. Senoglu and C.H. Aladag, *Statistical determination of significant particle swarm optimization parameters: the case of Weibull distribution*, Soft Comput. **26**, 12623-12634, 2022.
[3] H.M. Alshamlan, G.H. Badr and Y.A. Alohali, *Genetic Bee Colony (GBC) algorithm: A new gene selection method for microarray cancer classification*, Comput. Biol. Chem. **56**, 49-60, 2015.
[4] J.C. Bansal and K. Deep, *A modified binary particle swarm optimization for Knapsack problems*, Appl. Math. Comput. **22**, 11042-11061, 2012.
[5] R.E. Bellman, *Dynamic Programming*, Princeton University Press, Princeton, NJ, 1957.
[6] R.E. Bellman, *Dynamic Programming*, Science **153** (3731), 34-37, 1966.
[7] B. Bonev, F. Escolano, D. Giorgi and S. Biasotti, *Hybrid variable neighborhood search and simulated annealing algorithm to estimate the three parameters of the Weibull distribution*, Comput. Vision Image Understanding **117** (3), 214-228, 2013.
[8] R. Brits, A.P. Engelbrecht and F. Van Den Bergh, *A niching particle swarm optimizer*, Proceedings of the Conference on Simulated Evolution and Learning, 692-696, 2002.
[9] K. Chen, F.Y. Zhou and X.F. Yuan, *Hybrid particle swarm optimization with spiral-shaped mechanism for feature selection*, Expert Syst. Appl. **128**, 140-156, 2019.
[10] Y. Chen, J. Liu, J. Zhu and Z. Wang, *An improved binary particle swarm optimization combing V-shaped and U-shaped transfer function*, Evol. Intell. **16**, 1653-1666, 2023.
[11] L.Y. Chuang, C.H. Yang and J.C. Li, *Chaotic maps based on binary particle swarm optimization for feature selection*, Appl. Soft Comput. **11** (1), 239-248, 2011.
[12] M. Clerc and J. Kennedy, *The particle swarm - explosion, stability, and convergence in a multidimensional complex space*, IEEE Trans. Evol. Comput. **6** (1), 58-73, 2002.
[13] R.C. Eberhart and Y. Shi, *Particle swarm optimization: Developments, applications and resources*, Proceedings of the 2001 Congress on Evolutionary Computation **1**, 81-86, 2001.
[14] A.E. Eiben, R. Hinterding and Z. Michalewicz, *Parameter control in evolutionary algorithm*, IEEE Trans. Evol. Comput. **3** (2), 124-141, 1999.
[15] P.A. Estévez, M. Tesmer, C.A. Perez and J.M. Zurada, *Normalized mutual information feature selection*, IEEE Trans. Neural Networks **20** (2), 189-201, 2009.
[16] A.A. Ewees, M.A. El Aziz and A.E. Hassanien, *Chaotic multi-verse optimizer-based feature selection*, Neural Comput. Appl. **31**, 991-1006, 2019.
[17] A.J. Ferreira and M.A.T. Figueiredo, *An unsupervised approach to feature discretization and selection*, Pattern Recognit. **45** (9), 3048-3060, 2012.
[18] R.A. Fisher, *The design of experiments*, Oliver and Boyd, 1935.
[19] Q. Gu, Z. Li and J. Han, *Generalized fisher score for feature selection*, UAI'11: Proceedings of the Twenty-Seventh Conference on Uncertainty in Artificial Intelligence, 266-273, 2011.
[20] S.S. Guo, J.S. Wang and M.W. Guo, *Z-Shaped Transfer Functions for Binary Particle Swarm Optimization Algorithm*, Comput. Intell. Neurosci. **2020**, 6502807, 2020.

[21] X. He, D. Cai and P. Niyogi, *Laplacian Score for feature selection*, NIPS'05: Proceedings of the 18th International Conference on Neural Information Processing Systems **18**, 507-514, 2005.

[22] A.E. Hegazy, M.A. Makhlouf and G.S. El-Tawel, *Feature Selection Using Chaotic Salp Swarm Algorithm for Data Classification*, Arabian J. Sci. Eng. **44**, 3801-3816, 2019.

[23] C.L. Huang and J.F. Dun, *A distributed PSOSVM hybrid system with feature selection and parameter optimization*, Appl. Soft Comput. **8** (4), 1381-1391, 2008.

[24] M.J. Islam, X. Li and Y. Mei, *A time-varying transfer function for balancing the exploration and exploitation ability of a binary PSO*, Appl. Soft Comput. **59**, 182-196, 2017.

[25] I. Jain, V.K. Jain and R. Jain, *Correlation feature selection based improved-Binary Particle Swarm Optimization for gene selection and cancer classification*, Appl. Soft Comput. **62**, 203-215, 2018.

[26] H. Jiang, C. Kwong, Z. Chen and Y.C. Ysim, *Chaos particle swarm optimization and TS fuzzy modeling approaches to constrained predictive control*, Expert Syst. Appl. **39** (1), 194-201, 2012.

[27] J. Kennedy and R. Eberhart, *Particle swarm optimization*, Proceedings of ICNN'95 - International Conference on Neural Networks **4**, 1942-1948, 1995.

[28] J. Kennedy and R.C. Eberhart, *A discrete binary version of the Particle Swarm Optimization*, 1997 IEEE International Conference on Systems, Man, and Cybernetics. Computational Cybernetics and Simulation **5**, 4104-4108, 1997.

[29] H. Khosravi, B. Amiri, N. Yazdanjue and V. Babaiyan, *An improved group teaching optimization algorithm based on local search and chaotic map for feature selection in high-dimensional data*, Expert Syst. Appl. **204**, 117493, 2022.

[30] M. Labani, P. Moradi, F. Ahmadizar and M. Jalili, *A novel multivariate filter method for feature selection in text classification problems*, Eng. Appl. Artif. Intell. **70**, 25-37, 2018.

[31] S. Lee, S. Soak, S. Oh, W. Pedrycz and M. Jeon, *Modified binary particle swarm optimization*, Prog. Nat. Sci. **18** (9), 1161-1166, 2008.

[32] M. Mafarja, I. Aljarah, A.A. Heidari, H. Faris, P. Fournier-Viger, X. Li and S. Mirjalili, *Binary dragonfly optimization for feature selection using time-varying transfer functions*, Knowledge-Based Syst. **161**, 185-204, 2018.

[33] S. Mirjalili and A. Lewis, *S-shaped versus V-shaped transfer functions for binary Particle Swarm Optimization*, Swarm Evol. Comput. **9**, 1-14, 2013.

[34] S. Mirjalili, H. Zhang, S. Mirjalili, S. Chalup and N. Nomani, *A Novel U-Shaped Transfer Function for Binary Particle Swarm Optimisation*, In Soft Computing for Problem Solving 2019: Proceedings of SocProS 2019 **1**, 241-259, 2020.

[35] D.C. Montgomery, *Design and analysis of experiments*, Wiley, New Jersey, 2013.

[36] P. Moradi and M. Gholampour, *A hybrid particle swarm optimization for feature subset selection by integrating a novel local search strategy*, Appl. Soft Comput. **43**, 117-130, 2016.

[37] P. Moradi and M. Gholampour, *A novel particle swarm optimization algorithm with adaptive inertia weight*, Appl. Soft Comput. **11** (4), 3658-3670, 2011.

[38] H. Peng, F. Long and C. Ding, *Feature selection based on mutual information: Criteria of Max-Dependency, Max-Relevance, and Min-Redundancy*, IEEE Trans. Pattern Anal. Mach. Intell. **27** (8), 1226-1238, 2005.

[39] Y. Prasad, K.K. Biswas and M. Hanmandlu, *A recursive PSO scheme for gene selection in microarray data*, Appl. Soft Comput. **71**, 213-225, 2018.

[40] O.S. Qasim and Z.Y. Algamal, *Feature selection using particle swarm optimization-based logistic regression model*, Chemom. Intell. Lab. Syst. **182**, 41-46, 2018.

[41] O.S. Qasim, N.A. Al-Thanoon and Z.Y. Algamal, *Feature selection based on chaotic binary black hole algorithm for data classification*, Chemom. Intell. Lab. Syst. **204**, 104104, 2020.

[42] L.E. Raileanu and K. Stoffel, *Theoretical comparison between the Gini Index and Information Gain criteria*, Ann. Math. Artif. Intell. **41**, 77-93, 2004.

[43] E. Rashedi, H. Nezamabadi-Pour and S. Saryazdi, *A simultaneous feature adaptation and feature selection method for content-based image retrieval systems*, Knowledge-Based Syst. **39**, 85-94, 2013.

[44] M. Rostami, S. Forouzandeh, K. Berahmand and M. Soltani, *Integration of multi-objective PSO based feature selection and node centrality for medical datasets*, Genomics **112** (6), 4370-4384, 2020.

[45] M. Rostami, K. Berahmand, E. Nasiri and S. Forouzande, *Review of swarm intelligence-based feature selection methods*, Eng. Appl. Artif. Intell. **100**, 104210, 2021.

[46] S. Saremi, S. Mirjalili and A. Lewis, *Biogeography-based optimisation with chaos*, Neural Comput. Appl. **25**, 1077-1097, 2014.

[47] G.I. Sayed, A. Darwish and A.E. Hassanien, *A New Chaotic Whale Optimization Algorithm for Features Selection*, J. Classif. **35**, 300-344, 2018.

[48] G.I. Sayed, G. Khoriba and M.H. Haggag, *A novel chaotic salp swarm algorithm for global optimization and feature selection*, Applied Intelligence **48**, 3462-3481, 2018.

[49] G.I. Sayed, A. Tharwat and A.E. Hassanien, *Chaotic dragonfly algorithm: an improved metaheuristic algorithm for feature selection*, Appl. Intell. **49**, 188-205, 2019.

[50] M. Shafipour, A. Rashno and S. Fadaeii, *Particle distance rank feature selection by particle swarm optimization*, Expert Syst. Appl. **185**, 115620, 2021.

[51] Y. Shi and R. Eberhart, *A modified particle swarm optimizer*, 1998 IEEE International Conference on Evolutionary Computation Proceedings. IEEE World Congress on Computational Intelligence, 69-73, 1998.

[52] U. Singh and S.N. Singh, *A new optimal feature selection scheme for classification of power quality disturbances based on ant colony framework*, Appl. Soft Comput. **74**, 216-225, 2019.

[53] M. Tahir, A. Tubaishat, F. Al-Obeidat, B. Shah, Z. Halim and M. Waqas, *A novel binary chaotic genetic algorithm for feature selection and its utility in affective computing and healthcare*, Neural Comput. Appl. **34**, 11453-11474, 2022.

[54] X. Tang, Y. Dai and Y. Xiang, *Feature selection based on feature interactions with application to text categorization*, Expert Syst. Appl. **120**, 207-216, 2019.

[55] A. Tharwat and A.E. Hassanien, *Chaotic antlion algorithm for parameter optimization of support vector machine*, Appl. Intell. **48**, 670-686, 2018.

[56] T.O. Ting, M.V.C. Rao and C.K. Loo, *A novel approach for unit commitment problem via an effective hybrid particle swarm optimization*, IEEE Trans. Power Syst. **21** (1), 411-418, 2006.

[57] J. Too and A.R. Abdullah, *Chaotic Atom Search Optimization for Feature Selection*, Arabian J. Sci. Eng. **45**, 6063-6079, 2020.

[58] A. M. Unler and R.B. Chinnam, *mr2PSO: A maximum relevance minimum redundancy feature selection method based on swarm intelligence for support vector machine classification*, Inf. Sci. **181** (20), 4625-4641, 2011.

[59] F. Van Den Bergh and A.P. Engelbrecht, *Effects of swarm size on cooperative particle swarm optimisers*, GECCO'01: Proceedings of the 3rd Annual Conference on Genetic and Evolutionary Computation, 892-899, 2001.

[60] V.N. Vapnik, *The Nature of Statistical Learning Theory*, Springer, 1995.

[61] L. Wang, X. Wang, J. Fu and L. Zhen, *A novel probability binary particle swarm optimization algorithm and its application*, J. Softw. **3** (9), 28-35, 2008.

[62] H. Wang, Y. Zhang, J. Zhang, T. Li and L. Pengg, *A factor graph model for unsupervised feature selection*, Inf. Sci. **480**, 144-159, 2019.

[63] R.A. Welikala, M.M. Fraz, J. Dehmeshki, A. Hoppe, V. Tah, S. Mann, T.H. Williamson and S.A. Barman, *Genetic algorithm based feature selection combined with dual classification for the automated detection of proliferative diabetic retinopathy*, Comput. Med. Imaging Graphics **43**, 64-77, 2015.

[64] Y. Xu, G. Jones, J. Li, B. Wang and C. Sun, *Study on mutual information-based feature selection for text categorization*, J.Comput. Inf. Syst. **3** (3), 1007-1012, 2007.

[65] X. Xu, H. Rong, M. Trovati, M. Liptrott and N. Bessis, *CS-PSO: chaotic particle swarm optimization algorithm for solving combinatorial optimization problems*, Soft Comput. **22**, 783-795, 2018.

[66] B. Xue, M. Zhang and W.N. Browne, *Particle swarm optimization for feature selection in classification: A multi-objective approach*, IEEE Trans. Cybern. **43** (6), 1656-1671, 2013.

[67] B. Xue, M. Zhang and W.N. Browne, *Particle swarm optimisation for feature selection in classification: Novel initialisation and updating mechanisms*, Appl. Soft Comput. **18**, 261-276, 2014.

[68] B. Xue, M. Zhang, W.N. Browne and X. Yao, *A Survey on Evolutionary Computation Approaches to Feature Selection*, IEEE Trans. Evol. Comput. **20** (4), 606-626, 2016.

[69] Y. Xue, T. Tang, W. Pang and A.X. Liu, *Self-adaptive parameter and strategy based particle swarm optimization for large-scale feature selection problems with multiple classifiers*, Appl. Soft Comput. **88**, 106031, 2020.

[70] M. Yamada, W. Jitkrittum, L. Sigal, E.P. Xing and M. Sugiyama, *High-Dimensional feature selection by feature-Wise kernelized lasso*, Neural Comput. **26** (1), 185-207, 2014.

[71] F. Yates, *The design and analysis of factorial experiments*, Imperial Bureau of Soil Science, 1937.