HYBRID BINARY WHALE OPTIMIZATION ALGORITHM BASED ON TAPER SHAPED TRANSFER FUNCTION FOR SOFTWARE DEFECT PREDICTION

Zakaria A. Hamed Alnaish, Safwan O. Hasoon

Abstract: Reliability is one of the key factors used to gauge software quality. Software defect prediction (SDP) is one of the most important factors which affects measuring software’s reliability. Additionally, the high dimensionality of the features has a direct effect on the accuracy of SDP models. The objective of this paper is to propose a hybrid binary whale optimization algorithm (BWOA) based on taper-shape transfer functions for solving feature selection problems and dimension reduction with a KNN classifier as a new software defect prediction method. In this paper, the values of a real vector that represents the individual encoding have been converted to binary vector by using the four types of Taper-shaped transfer functions to enhance the performance of the BWOA to reduce the dimension of the search space. The performance of the suggested method (T-BWOA-KNN) was evaluated using ten standard software defect prediction datasets from the PROMISE and NASA repositories depending on the K-Nearest Neighbor (KNN) classifier. Seven evaluation metrics have been used to assess the effectiveness of the suggested method. The experimental results have shown that the performance of T-BWOA-KNN produced promising results compared to other methods including ten methods from the literature, four types of T-BWOA with the KNN classifier. In addition, the obtained results are compared and analyzed with other methods from the literature in terms of the average number of selected features (SF) and accuracy rate (ACC) using the Kendall W test. In this paper, a new hybrid software defect prediction method called T-BWOA-KNN has been proposed which is concerned with the feature selection problem. The experimental results have proved that T-BWOA-KNN produced promising performance compared with other methods for most datasets.

Keywords: feature selection, binary whale optimization algorithm, taper-shaped transfer function, software defect prediction

HYBRIDOWY, BINARNY ALGORYTM WOA OPARTY NA TRANSMITANCJI STOŻKOWEJ DO PROZNOZOWANIA DEFEKTÓW OPROGRAMOWANIA


Słowa kluczowe: wybór cech, algorytm optymalizacji binarnej, transmitancja stożkowa, przewidywanie defektów oprogramowania

Introduction

Software defect prediction (SDP) is a crucial software quality assurance technique, which could extract defects of any software and help the developers or maintainers efficiently detect the potentially defective modules [30]. To minimize the undesirable effects of any software, software defect prediction should be done before delivering the software to the customers [26]. Generally, building efficient software defect prediction models depends on soft computing (SC), machine learning (ML) [21], software features (metrics) that are generated in the software development process, or code complexity [28]. Besides that, some of these features are unrelated and jobless features.

Consequently, machine learning (ML) techniques consider a good solution for building software defect prediction modules in software projects. Additionally, numerous types of research have confirmed the effectiveness of machine learning (ML) techniques in enhancing the efficiency of software defect prediction methods. The common techniques which are utilized in SDP are SVM, DT, NB, LR, ANN, DF, and CNN [5, 23, 29].

Concurrently with software development and because all life applications are managed by computer systems, and the volume of the produced data will become huge. Thus, the basic ML techniques became unpractical in some fields and need improvements, especially in the SDP field. Dimension reduction considers one of the common methods for enhancing the machine-learning performance of the input data [11]. One of the main techniques of dimension reduction is feature selection (FS) [7, 10].

The feature selection process (FS) involves identifying and choosing the best relevant features for any problem domain to achieve the highest accuracy [7]. The feature selection algorithms are categorized into types [1, 9]. The first type called the filter method, in which the feature selection process does not involve the classifier. The second type called the wrapper method, in which the feature selection relies on the used classifier, which serves as an effective evaluation criterion for choosing the best features. The hybrid method combines wrapper and filter techniques to select the subset of features that rely on the classifier’s design [1, 9, 10].

Recently, there are many variants of metaheuristic algorithms that have been implemented for solving feature selection problems in SDP. For instance, in [3] the authors proposed to use fourteen filters as subset feature selection (SFS) methods and four filter feature ranking (FFR). The proposed methods evaluated by utilizing four classifiers based on five types of software defect datasets. The paper compares the performance of feature selection methods, including Relief, Chi-Square, Information Gain, and Correlation-based Feature Selection (CFS). Their results have shown that the efficiency of each method varies according to the dataset type and prediction model.

In [16], the authors proposed a classification framework based on Multi-Layer Perceptron (MLP) and many filter feature selection techniques to predict software defects. The suggested...
methods are implemented with and without oversampling techniques for manipulating data misbalancing. The authors have used Twelve datasets with four evaluation metrics. The produced results have demonstrated that the framework with class balancing produces good performance with all datasets.

In [26], the authors proposed a novel approach to improve the performance of a layered-recurrent neural network (L-RNN) for software fault prediction. By using feature selection techniques, the authors aim to eliminate irrelevant features and improve the accuracy of fault prediction. The authors employ three different wrapper feature selection algorithms (Binary Genetic Algorithm, Binary Particle Swarm Optimization, and Binary Ant Colony Optimization) iteratively to select the most important software metrics. The results of the experiments, which are conducted on nineteen common datasets from the PROMISE repository, have shown that the proposed approach achieves an excellent classification rate and outperforms existing results found in the literature. Therefore, the authors claimed that feature selection plays a vital role in enhancing the performance of the layered recurrent neural network for software fault prediction.

In [6], to address the high dimensionality and filter rank selection problem in software defect prediction, the authors suggested a unique rank aggregation-based multilifier FS method. The suggested approaches combine rank lists produced by various filter methods into a single aggregated rank list by employing rank aggregation algorithms. On nine defect datasets from the NASA repository, the efficiency of the suggested technique was assessed using Decision Tree (DT) and Naive Bayes (NB) models. According to the experimental findings, the proposed methods had a greater impact on the prediction performances of NB and DT models than other FS methods.

In [25], a new version of the Binary Moth Flame Optimization called (EBMFO algorithm) and Adaptive synthetic sampling (ADASYN) has been proposed to predict software defects. The algorithm addresses the issue of imbalanced data distribution and enhances the input dataset for accurate predictions. The proposed EBMFO algorithm is employed as a wrapper feature selection algorithm, which selects the most relevant features from the input dataset to enhance the overall performance of classifiers. Also, the authors demonstrated that transfer functions are important in the Enhanced Binary Moth Flame Optimization algorithm as they are used to convert the continuous search space to binary search space. The remaining whales then strive to adjust their positions to the desired prey or positioned in close proximity to the optimal solution. The given results show that the suggested EBMFO improves the classifier’s efficiency overall and outperforms the findings in the literature.

In [11], the authors suggested an enhanced version of the Whale Optimization Algorithm (WOA) by incorporating natural selection operators to improve software fault prediction. The authors introduce two natural selection operators: crossover and mutation. The crossover operator facilitates the exchange of genetic information between two whale individuals, while the mutation operator introduces small changes to the solution space. The proposed Boosted Whale Optimization Algorithm (BWOA) is evaluated using multiple real-world software datasets. The experimental results demonstrate that BWOA outperforms traditional WOA and other state-of-the-art fault prediction techniques in terms of accuracy, precision, recall, and F-measure. Thus, the BWOA algorithm considers an efficient method for solving feature selection problems in SDP.

In light of the studies mentioned above, the motivation for this study is raised. Although there were many methods have been proposed recently for solving feature selection problems in SDP, there is still a need to develop and enhance a robust feature selection method because not all the existing feature selection techniques produce the best accuracy in solving SDP with all datasets. Since the transfer functions dynamically adjust the search behaviour based on the fitness values of the solutions and allow for more efficient exploration of the search space [12], Thus, using an appropriate transfer function to convert the continuous search space to binary search space considers a big challenge. So, introducing a new transfer function considers the objective of this study.

Thus, this paper proposes a new robust feature selection method based on BWOA and taper-shaped transfer function. The following points summarized the contributions of this paper:

1) Propose a new hybrid feature selection method using a binary whale optimization algorithm and Taper Shaped Transfer Function to get the most significant features for solving software defect prediction problems.

2) The proposed method (T-BWOA-KNN) has been evaluated based on eleven SDP datasets with (KNN) classifier in terms of accuracy rate, average of selected features number, sensitivity, specificity, accuracy, G-mean, and error rate (ER)

3) Kandell W test has been implemented to show the significant difference for the performance of the (T-BWOA-KNN) and rank it with the techniques from the literature.

The rest of this paper is arranged as follows: section two includes an explanation of the binary whale optimization algorithm (BWOA). Section three explains the suggested method. The experiment setup has been explained in section four. Section five includes the experimental results and discussions. Section six includes a comparison performance of the proposed T-BWOA-KNN with other methods from the literature. Finally, section seven includes the conclusion.

1. Binary Whale Optimization Algorithm (BWOA)

First, Whale Optimization Algorithm (WOA) [20] is an optimization algorithm that is inspired by the foraging behavior of whales. Whales exhibit a feeding technique known as the bubble-net method during their foraging activities. Nevertheless, within the Whales Optimization Algorithm, the instant most promising solution is designated as either the desired prey or positioned in close proximity to the optimal solution. The remaining whales then strive to adjust their positions towards this superior choice. For solving any optimization problem, the implementation of the WOA algorithm could be represented as (n, d) represents the number of Whales, and x^* represents the value for the position of the (i)th whale at iteration (t). For instance, assume at iteration t, for n whales and d dimensions, the whales are denoted as shown in matrix (X):

\[
\text{whereas } (X) = \begin{bmatrix} x^*_1 & x^*_2 & \ldots & x^*_d \\ x_1^* & x_2^* & \ldots & x_d^* \\ \vdots & \vdots & \ddots & \vdots \\ x_1^n & x_2^n & \ldots & x_d^n \end{bmatrix}
\]

Each row in the matrix (X) denotes one probable solution.

The mathematical simulation process of WOA swarming behavior could be described as shown in the following equations:

\[
D = |c.X(t) - X(t)|
\]

\[
X(t + 1) = X(t + 1) - a.D
\]

where i represents the iteration number, X represents the position vector, and X' represents the position vector of the best-founded solution. A and C represent the coefficient vectors as shown in equations 3 and 4, respectively:

\[
A = 2ar - a
\]

\[
C = 2r
\]

The variable "a" is assigned a random value within the range of [0, 1], while the variable "a" undergoes a linear decrease from two to zero throughout the repeated cycles. Similar to other optimization algorithms, this algorithm consists of two main phases: exploration and exploitation. The exploitation phase involves two processes:

1. Shrinking encircling mechanism: This mechanism is achieved by reducing the value of "a" based on equation 4. It is important to note that "a" is a random value within the range of [0, 1].

2. Spiral updating position: this process includes calculating the distance between the whale and the prey. Then, the spiral
The proposed method (T-BWOA) for software defect prediction

In the binary Whale Optimization Algorithm (BWOA), the solutions are represented by binary space [7]. In feature selection problem, the continuous spaces should be transforming to the equivalent binary space (i.e. 0 or 1), where the value 0 means that the feature is irrelevant and the value 1 means that the feature is relevant.

Also, in the BWOA, the transformation process needed using a appropriate transfer function such as sigmoid function (S-shaped) and hyperbolic function (V-shaped). The transfer function plays an important role in the performance of the binary optimization algorithm (BOA) [12]. Hence, the motivation of this study is to propose a new hybrid BWOA called taper shaped transfer function-binary whale optimization algorithm (T-BWOA) for solving feature selection problem in software defect prediction. In this study, four taper-shaped transfer functions (T1-T4) [12] are used to convert the continues values (x) to real number within [0,1] based on the equations (9-12):

\[
T1(x) = \frac{1}{1 + e^{-ax}}
\]

(9)

\[
T2(x) = \frac{1}{1 + e^{-ax}}
\]

(10)

\[
T3(x) = \frac{1}{1 + e^{-ax}}
\]

(11)

\[
T4(x) = \frac{1}{1 + e^{-ax}}
\]

(12)

Figure (1) shows the curves of Taper-shaped transfer functions.

Fig. 1. The curves of Taper-shaped transfer functions [12]

Then, convert the obtained values by \( Tk(x)(k = 1, 2, 3, 4) \), to the binary space based on equation (13)

\[
x_b = \begin{cases} 
1, & \text{if } t_b(x) \leq T_k \\
0, & \text{otherwise}
\end{cases}
\]

(13)

Then, the fitness value is calculated using a fitness function which is represented by equation (14)

\[
f(x(t_{1})) = \frac{1}{\pi + \beta |x'|}
\]

(14)

where \( C \) represents the error rate of the classification, \( X \) represents the number of selected features by whale \( x(t_{1}) \). \( K \) represents all the features, \( \alpha = 0.09 \) and \( \beta = 1 - \alpha \). Algorithm 1 shows the steps of the proposed T-BWOA.

Algorithm 1: (T-BWOA-KNN)
1: start
2: The inputs: a number of whales (N) and maximum number of iterations (t).
3: The output: the best whale’s positions
4: set the Initial values of a and N
5: compute the fitness value for the whales according to equation (4) and find the best search agent (X’)
6: while stop condition (maximum iteration (t)) has not satisfied do
7: for i = 1 : N
8: Calculate and Update the following parameters:
9: \( a = 2 \cdot t \cdot (2t) \)
10: \( A = 2 \cdot a \cdot \text{rand}() - a \)
11: \( C = 2 \cdot \text{rand}() \)
12: \( P = \text{rand()}, \)
13: \( L = \text{random number in [-1,1]} \)
14: if \( p > 0.5 \) then
15: \( x(t + 1) = D' + e^{b \cdot t} \cdot \cos(2m) + x'(t) \)
16: else (p<0.5) do
17: if \( |A| < 1 \) then
18: \( x(t + 1) = x'(t) - A + D \)
19: \( D = |C + x'(t) - x(t)| \)
20: else \( |A| \geq 1 \) do
21: \( x(t + 1) = x'(t) - A + D \)
22: \( D = |C + x'(t) - x(t)| \)
23: \( End \) if
24: \( End \) if
25: \( End \) if
26: Convert \( x(t + 1) \) to binary space using four types of taper shaped transfer functions based on equation (13) for each t value
27: End for
28: calculate the fitness value using KNN classifier for each whale based on equation (14)
29: Update the value of \( x' \), if there is a better solution
30: t := t+1
31: end while

Also, figure 2. demonstrates the framework of the suggested T-BWOA-KNN.

Fig. 2. The framework of the proposed T-BWOA-KNN

3. Experiments setup

In this section, the datasets description, the pre-processing, performance evaluation metrics, parameters configuration, and the used statistical test are explained in details in the following subsections. The results of each experiment are obtained by running each experiment ten times separately then calculate the average of all experiments.
3.1. Datasets description

To measure the performance of the proposed method (T-BWOA-KNN), eleven software datasets were used which are available free in NASA and PROMISE repository [2, 17, 22]. Each dataset includes different number of records. Each record represents one project. Also, each project has many features (attribute).

Each dataset has divided into two groups which are: training part consists of 70% of the total size of the dataset, while the remaining instances form the testing part. Each experiment is independently conducted 10 times. Table 1 shows a summary of the used dataset.

### Table 1. Datasets description

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Attribute (features)</th>
<th>No. of records</th>
<th>Defective records</th>
<th>Non-defective records</th>
</tr>
</thead>
<tbody>
<tr>
<td>CM1</td>
<td>37</td>
<td>327</td>
<td>42</td>
<td>285</td>
</tr>
<tr>
<td>JM1</td>
<td>37</td>
<td>327</td>
<td>42</td>
<td>285</td>
</tr>
<tr>
<td>KC1</td>
<td>37</td>
<td>327</td>
<td>42</td>
<td>285</td>
</tr>
<tr>
<td>KC2</td>
<td>37</td>
<td>327</td>
<td>42</td>
<td>285</td>
</tr>
<tr>
<td>MW1</td>
<td>37</td>
<td>327</td>
<td>42</td>
<td>285</td>
</tr>
<tr>
<td>PC1</td>
<td>37</td>
<td>327</td>
<td>42</td>
<td>285</td>
</tr>
<tr>
<td>PC2</td>
<td>37</td>
<td>327</td>
<td>42</td>
<td>285</td>
</tr>
<tr>
<td>PC3</td>
<td>37</td>
<td>327</td>
<td>42</td>
<td>285</td>
</tr>
<tr>
<td>PC4</td>
<td>37</td>
<td>327</td>
<td>42</td>
<td>285</td>
</tr>
<tr>
<td>PC5</td>
<td>37</td>
<td>327</td>
<td>42</td>
<td>285</td>
</tr>
<tr>
<td>Tomcat</td>
<td>20</td>
<td>858</td>
<td>77</td>
<td>781</td>
</tr>
</tbody>
</table>

3.2. Pre-processing

Min-Max scaler normalization has been applied in this paper because it is a normalization technique commonly used in machine learning and data preprocessing. It scales the features of a dataset to a specific range, typically between 0 and 1 [27], as shown in equation (15).

\[
\text{Normalized value (x')} = \frac{(x - \text{min})}{(\text{max} - \text{min})} \quad (15)
\]

3.3. Performance evaluation

Predicting the defective classes in a target version based on a confusion matrix, as shown in table 2.

### Table 2. Confusion matrix

<table>
<thead>
<tr>
<th>Predicted output</th>
<th>Actual output</th>
<th>Positive</th>
<th>Negative</th>
</tr>
</thead>
<tbody>
<tr>
<td>Positive</td>
<td>TP</td>
<td>FP</td>
<td></td>
</tr>
<tr>
<td>Negative</td>
<td>FN</td>
<td>TN</td>
<td></td>
</tr>
</tbody>
</table>

Where: TP is true positive, TN is true negative, FP is false positive, FN is false negative. In this research, based on the values of these four indicators, seven evaluation metrics were calculated to measure the performance of the suggested method which are: classification accuracy (ACC), area under curve (AUC), sensitivity (SN), specificity (SP), number of selected features (SF), error rate (ER), and G-mean.

The classification accuracy (ACC) represents the percentage of the instances that have been classified correctly [30], it is calculated using equation (16).

\[
\text{ACC} = \frac{(TP + TN)}{(TP + TN + FP + FN)} \quad (16)
\]

Also, area under curve (AUC) has been used to assess the distinguishing ability of the proposed model. Its value falls within [0,1], the higher the better. In addition, AUC is appropriate for evaluating class-imbalanced datasets [8]. It is calculated based on equation (17)

\[
\text{AUC}=\frac{1+(TPR-FPR)\times 2}{1+(TPR-FPR)} \quad (17)
\]

where TPR is the proportion of positive label instances that were predicted correctly [21], as shown in equation (18).

\[
\text{TPR}=\frac{TP}{(TP+FN)} \quad (18)
\]

And, false positive rate (FPR) represents the proportion of negative instances that are incorrectly predicted as positive by the model, which is calculated as shown in equation (19).

\[
\text{FPR}=\frac{FP}{(FP+TN)} \quad (19)
\]

The score’s range of the Kendall W test will be within the interval [0,1] and the decision will depend on the following roles[28]:

- 0.00 <= \( w \) < 0.20 – Slight agreement
- 0.20 <= \( w \) < 0.40 – Fair agreement
- 0.40 <= \( w \) < 0.60 – Moderate agreement
- 0.60 <= \( w \) < 0.80 – Substantial agreement
- \( w \) >= 0.80 – Almost perfect agreement

3.5. Statistical test

In this research, Kendall W test has been used to show the significant performance of the suggested techniques and rank it with other SDP methods from the literature. The Kendall W test is a statistical test used to measure the degree of agreement among multiple observers or raters. It assesses the extent to which the rankings or ratings assigned by different observers to a set of items or subjects are consistent. In the Kendall W test, each case represents a judge or rater, while each variable represents the thing or person being assessed, below are the steps for calculating the Kendall W test score [19]:

1. Assume the object (i) is considered as the SDP method, (ranked objects) is given the rank rij by the raters j (datasets), where there are in total (n) objects and (m) raters. The total rank (R) given to object (i) is calculated using equation (24)

\[
R_i = \sum_j^n r_{ij} \quad (24)
\]

2. The average value of (R) is calculated using equation (25)

\[
\bar{R} = \frac{1}{n} \sum_{i=1}^n R_i \quad (25)
\]

3. The sum of squared deviations, S, is calculated using equation (26)

\[
S = \sum_{i=1}^n (R_i - \bar{R})^2 \quad (26)
\]

4. The Kendall’s W coefficient is calculated using equation (27)

\[
W = \frac{12S}{m(n^2-n)} \quad (27)
\]

In addition, the error rate (ER) has been calculated based on equation (23), which shows the misclassification rate for the classes.

\[
\text{ER}=1-\frac{\text{ACC}}{} \quad (23)
\]
4. The experiments results and discussions

The final implementation of the proposed T-BWOA with KNN classifier is done in Python 3.9.7 using Spyder which is served as the development environment. The experimental results for the eleven datasets are provided in tables 4–10. These tables show the obtained results for the suggested method (T-BWOA-KNN) for solving SDF problem in terms of accuracy rate (ACC), the average number of selected features (SF), sensitivity (SN), specificity (SP), area under curve (AUC), G-mean, and error rate (ER). Every table shows the values of one evaluation metric and each row in the table displays the obtained result for one dataset, with the best result shaded for each method whereas the first column represents name of the dataset and the rest represents the proposed methods.

In terms of accuracy, as shown in table 4, T2-BWOA has produced the highest accuracy for JM1, KC1, PC2, PC3 datasets. Meanwhile, T1-BWOA has produced the highest accuracy for CM1, JM1, PC1 datasets. T3-BWOA has produced the highest accuracy for MW1, PC4, TOMCAT datasets. T4-BWOA has produced the highest accuracy for JM1, KC3, PC5 datasets. In summary, T2-BWOA has the best performance in terms of accuracy for four from eleven datasets comparing with others which has the best performance for three datasets only which are shaded by gray color, although T1-BWOA and T3-BWOA has produced the best performance in terms of accuracy mean (0.891%).

In terms of sensitivity, as shown in table 5, T1-BWOA, and T3-BWOA have produced the highest sensitivity for four different datasets which are shaded by gray colour. Meanwhile, T4-BWOA has produced the highest sensitivity for only two datasets. At the same time, T2-BWOA has produced the highest performance in terms of sensitivity mean. Although T2-BWOA has the best performance in terms of sensitivity for three datasets from eleven, the T2-BWOA has produced the best performance in terms of sensitivity mean (0.296%).

In terms of Specificity, as shown in table 6, T2-BWOA has produced the highest specificity for five datasets from eleven. Meanwhile, T1-BWOA has produced the highest specificity for PC1 dataset only. T3-BWOA has produced the highest specificity for four datasets. T4-BWOA has produced the highest specificity for three datasets which are shaded by gray color. In summary, T2-BWOA has the best performance in terms of specificity for five datasets from eleven comparing with others. Also, the T2-BWOA has produced the best performance in terms of specificity mean (0.964%).

In terms of AUC, as shown in table 8, T4-BWOA has produced the highest AUC for four datasets from eleven. Meanwhile, T1-BWOA has produced the highest AUC for three datasets. T2-BWOA has produced the highest AUC for PC2 dataset only. T3-BWOA has produced the highest accuracy for three datasets. In summary, T4-BWOA has the best performance in terms of AUC for four datasets from eleven datasets compared with others which are shaded by gray color. Also, the T4-BWOA has produced the best performance in terms of AUC mean (0.6257%).

In terms of error rate, as shown in table 9, T2-BWOA has produced the minimum error rate for four datasets from eleven. Meanwhile, T1-BWOA, T3-BWOA, and T4-BWOA has produced the minimum error rate for three datasets which are shaded in gray colour. Although, T3-BWOA has produced the best performance in terms of mean error rate (0.1080%).
Among the six comparatives above, four cases clarified that T2-BWOA had produced the best performance for solving feature selection in SDP compared with the others.

For more proof of the effectiveness of the suggested method (T-BWOA-KNN) in selecting the optimal features that yield the maximum classification accuracy, Table 10 shows the number of best features for each method on average. The smaller number of features means that the performance of the method is better. As shown in Table 10, T2-BWOA has fewer features for nine datasets from eleven compared to other methods. For example, for CM1 dataset, T2-BWOA selected around two features compared to three, four, four, and four features for T1-BWOA-KNN, T3-BWOA-KNN, and T4-BWOA-KNN, respectively. Thus, the proposed T2-BWOA has the minimum number of the selected features with the most of the datasets by comparing it with other methods. So, it is considered the best method compared with others for solving feature selection problem in SDP.

For more clarity of the experiment results, a boxplot is used for all datasets in terms of accuracy to show the minimum, median, and maximum accuracy of the obtained results for running the experiments ten times autonomously where the red line for each block of the dataset represents the median. As shown in Fig. 3, the T2-BWOA-KNN method achieved the highest values for the PC2 dataset. On the other hand, it achieved the worst values for the KC2 dataset. At the same time, it produces the best performance in terms of accuracy for four datasets from eleven compared to other methods that produced the best accuracy for three datasets only.

5. Comparison of T-BWOA-KNN with other methods from the literature

After determining that T-BWOA-KNN has produced the best results for solving feature algorithms in SDP in terms of accuracy and minimum selected features based on the aforementioned results, the performance of T-BWOA-KNN has been compared with the performance of other researches from the literature that experimented on the same datasets in terms of accuracy and selected features.

Fig. 3. Boxplot of all datasets for the proposed method(T1~T4-BWOA-KNN) in terms of accuracy

The first research enhanced wrapper feature selection (EWFS) based on a dynamic re-ranking strategy by deploying two classifiers: Decision Tree (DT) and Naïve Bayes (NB) [5]; the second research used correlation-based feature subset selection (CFS) with metaheuristic algorithms such as GA, BAT, PSO, FS, and AS with multiple classifiers such as KNN [3]; the third research used rank aggregation which depends
on feature selection method with multifilters which is called (RMFFS) with two classifiers which are Decision Tree (DT) and Naïve Bayes (NB) [4]; the last research was proposing a hybrid method by using the multi-layer perceptron (MLP) with multi-filter feature selection technique called (MLP-FS) [16]. The comparison results are shown in table 11 and table 12, where” means that the value for this dataset is not available and the Shaded cells refer to the best-obtained results. It is clear that the performance of T-BWOA-KNN was better than other methods in most datasets in terms of accuracy and the number of selected features. For instance, the result in dataset PC5, an improvement of at least more than 30% in terms of accuracy can be achieved with T-BWOA-KNN as shown in table 11 and table 12, respectively.

In summary, the obtained results support the objective of this research that the taper-shaped transfer function can improve the overall performance of WOA with KNN classifiers. Last but not least, it can be concluded that T-BWOA-KNN is a useful tool for solving feature selection problems in SDP.

For more investigation and in order to show the significance of the proposed method (T-BWOA-KNN), the null hypothesis (H0) and alternative hypothesis (H1) are constructed as shown below:

H0: there is no significant difference between the accuracy of the proposed method (T-BWOA-KNN) with other methods from the literature.
H1: there is a significant difference between the accuracy of the proposed method (T-BWOA-KNN) with other methods from the literature. Thus, the Kendall W test has been implemented to show the significance and the ranking of the proposed method (T-BWOA-KNN) with other methods in the state-of-the-art that are used KNN classifier only in terms of the accuracy for mutual datasets (i.e. CM1, KC1, KC3, MW1) using Kendall W test results. The results of the Kendall W test for the mutual datasets indicate that the P-value (0.002) of the test is lower than α (0.05) which means that the condition of the null hypothesis, (H0) is rejected and the alternative hypothesis (H1) is accepted. This indicates that the proposed method (T-BWOA-KNN) is significant.

As shown in table 13, the performance of the proposed model (T-BWOA-KNN) is more significant than other methods in terms of accuracy for solving SDP based on p-value. Also, the overall ranking of all methods indicates that the proposed (T-BWOA-KNN) has the top ranks among other methods that used KNN classifier, as demonstrated in figure 4.

### Table 11. Comparison of T-BWOA-KNN to other methods in the state of art in terms of selected features

<table>
<thead>
<tr>
<th>Dataset</th>
<th>The Proposed method</th>
<th>[5]</th>
<th>[13]</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>T2_BWOA-KNN</td>
<td>NB+EWFS</td>
<td>DT+EWFS</td>
</tr>
<tr>
<td>CM1</td>
<td>2</td>
<td>4.7</td>
<td>7</td>
</tr>
<tr>
<td>JM1</td>
<td>2</td>
<td>7</td>
<td>7</td>
</tr>
<tr>
<td>KC1</td>
<td>4</td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td>KC3</td>
<td>4</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>MW1</td>
<td>6</td>
<td>5</td>
<td>6</td>
</tr>
<tr>
<td>PC1</td>
<td>3</td>
<td>5</td>
<td>6</td>
</tr>
<tr>
<td>PC2</td>
<td>2</td>
<td>3</td>
<td>5</td>
</tr>
<tr>
<td>PC3</td>
<td>2</td>
<td>3</td>
<td>5</td>
</tr>
<tr>
<td>PC4</td>
<td>5</td>
<td>6</td>
<td>6</td>
</tr>
<tr>
<td>PC5</td>
<td>2</td>
<td>3</td>
<td>6</td>
</tr>
<tr>
<td>TOMCAT</td>
<td>3</td>
<td>6</td>
<td>4</td>
</tr>
</tbody>
</table>

### Table 12. Comparison of T-BWOA-KNN to other methods in the state of art in terms of accuracy

<table>
<thead>
<tr>
<th>Dataset</th>
<th>proposed</th>
<th>[5]</th>
<th>[13]</th>
<th>[29]</th>
<th>[14]</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>T2+BWOA+</td>
<td>NB+EWFS</td>
<td>DT+EWFS</td>
<td>CFS+KNN+GA</td>
<td>CFS+KNN+BA</td>
</tr>
<tr>
<td>CM1</td>
<td>84.3</td>
<td>79.17</td>
<td>86.63</td>
<td>77.68</td>
<td>77.37</td>
</tr>
<tr>
<td>JM1</td>
<td>95.8</td>
<td>/</td>
<td>/</td>
<td>/</td>
<td>/</td>
</tr>
<tr>
<td>KC1</td>
<td>85.0</td>
<td>75.3</td>
<td>75.9</td>
<td>71.26</td>
<td>72.98</td>
</tr>
<tr>
<td>KC3</td>
<td>78.1</td>
<td>82.47</td>
<td>85.41</td>
<td>78.87</td>
<td>78.87</td>
</tr>
<tr>
<td>MW1</td>
<td>86.0</td>
<td>90.0</td>
<td>89.87</td>
<td>84</td>
<td>84.4</td>
</tr>
<tr>
<td>PC1</td>
<td>90.2</td>
<td>91.9</td>
<td>91.9</td>
<td>/</td>
<td>/</td>
</tr>
<tr>
<td>PC2</td>
<td>99.0</td>
<td>/</td>
<td>/</td>
<td>96.54</td>
<td>96.12</td>
</tr>
<tr>
<td>PC3</td>
<td>88.8</td>
<td>84.59</td>
<td>80.54</td>
<td>/</td>
<td>/</td>
</tr>
<tr>
<td>PC4</td>
<td>84.3</td>
<td>82.67</td>
<td>86.80</td>
<td>/</td>
<td>/</td>
</tr>
<tr>
<td>PC5</td>
<td>97.4</td>
<td>74.87</td>
<td>76.04</td>
<td>/</td>
<td>/</td>
</tr>
<tr>
<td>TOMCAT</td>
<td>89.2</td>
<td>90.96</td>
<td>92.02</td>
<td>/</td>
<td>/</td>
</tr>
</tbody>
</table>

### Table 13. Ranking of the proposed method (T-BWOA-KNN) with other methods using Kendall W test

<table>
<thead>
<tr>
<th>Mutual Datasets</th>
<th>W</th>
<th>P</th>
<th>Rank methods</th>
</tr>
</thead>
<tbody>
<tr>
<td>CM1, KC1, KC3</td>
<td>0.745</td>
<td>0.002</td>
<td>T1_BWOA_KNN 8.25</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>T3_BWOA_KNN 7.25</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>T4_BWOA_KNN 7.25</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>T2_BWOA_KNN 6.75</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>CFS_KNN BA 6.12</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>CFS_KNN GA 3.38</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>CFS_KNN PSO 3</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>CFS_KNN FS 2.5</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>CFS_KNN AS 2.5</td>
</tr>
</tbody>
</table>

Fig. 4. The ranking of (T-BWOA-KNN) with other methods from the literature.
6. Conclusion

This paper includes evolving the binary whale optimization algorithm (BWOA) by using the taper shaped transfer function to convert the continuous search space to binary search space for solving feature selection problems in SDP.

The main purpose of this research was to select the minimum number of relevant selected features with the highest accuracy for solving SDP.

The proposed method (T-BWOA-KNN) has been applied on eleven datasets which are CM1, JM1, KC1, KC3, MW1, PC1, PC2, PC3, PC4, PC5, TOMCAT that are obtained from NASA and promise repository. These datasets are varies based on the number of projects (patterns), attributes (features), and defect ratio. Each experiment has been used KNN classifier and repeated ten times autonomously to show the performance of the proposed method (T-BWOA-KNN) based on seven evaluations metrics.

The experimental results have shown that the proposed method T-BWOA-KNN has produced the highest classification accuracy and the minimum number of the selected features for most of the datasets compared to other methods. Also, it was showed that the performance of feature selection methods depends on dataset and the used classifier. In addition, the proposed method (T-BWOA-KNN) has the top ranks among other methods from the literature that used the mutual datasets and KNN classifier in terms of accuracy. As future works, utilizing another classifier such as a multilayer perceptron (MLP) or support vector machine (SVM) instead of the KNN. Also, suggest a new feature selection technique by hybridization Quasi-Oppositional Method with binary whale optimization algorithms as a feature selection method in SDP.

References