http://doi.org/10.35784/iapgos.4569

received: 5.09.2023 | revised: 20.10.2023 | accepted: 02.11.2023 | available online: 20.12.2023

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

**Zakaria A. Hamed Alnaish**[1,2], **Safwan O. Hasoon**[2]

[1]University of Mosul, College of Science, Mosul, Iraq, [2]University of Mosul, College of Computer Science and Mathematics, Mosul, Iraq

***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 BWOA to reduce the dimension of the search space. The performance of the suggested method (T-BWOA-KNN) was evaluated using eleven 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

# HYBRYDOWY, BINARNY ALGORYTM WOA OPARTY NA TRANSMITANCJI STOŻKOWEJ DO PROGNOZOWANIA DEFEKTÓW OPROGRAMOWANIA

***Streszczenie:*** *Niezawodność jest jednym z kluczowych czynników stosowanych do oceny jakości oprogramowania. Przewidywanie defektów oprogramowania SDP (ang. Software Defect Prediction) jest jednym z najważniejszych czynników wpływających na pomiar niezawodności oprogramowania. Dodatkowo, wysoka wymiarowość cech ma bezpośredni wpływ na dokładność modeli SDP. Celem artykułu jest zaproponowanie hybrydowego algorytmu optymalizacji BWOA (ang. Binary Whale Optimization Algorithm) w oparciu o transmitancję stożkową do rozwiązywania problemów selekcji cech i redukcji wymiarów za pomocą klasyfikatora KNN jako nowej metody przewidywania defektów oprogramowania. W artykule, wartości wektora rzeczywistego, reprezentującego indywidualne kodowanie zostały przekonwertowane na wektor binarny przy użyciu czterech typów funkcji transferu w kształcie stożka w celu zwiększenia wydajności BWOA i zmniejszenia wymiaru przestrzeni poszukiwań. Wydajność sugerowanej metody (T-BWOA-KNN) oceniano przy użyciu jedenastu standardowych zestawów danych do przewidywania defektów oprogramowania z repozytoriów PROMISE i NASA w zależności od klasyfikatora KNN. Do oceny skuteczności sugerowanej metody wykorzystano siedem wskaźników ewaluacyjnych. Wyniki eksperymentów wykazały, że działanie rozwiązania T-BWOA-KNN pozwoliło uzyskać obiecujące wyniki w porównaniu z innymi metodami, w tym dziesięcioma metodami na podstawie literatury, czterema typami T-BWOA z klasyfikatorem KNN. Dodatkowo, otrzymane wyniki zostały porównane i przeanalizowane innymi metodami z literatury pod kątem średniej liczby wybranych cech (SF) i współczynnika dokładności (ACC), z wykorzystaniem testu W. Kendalla. W pracy, zaproponowano nową hybrydową metodę przewidywania defektów oprogramowania, nazwaną T-BWOA-KNN, która dotyczy problemu wyboru cech. Wyniki eksperymentów wykazały, że w przypadku większości zbiorów danych T-BWOA-KNN uzyskała obiecującą wydajność w porównaniu z innymi metodami.*

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

## Introduction

Software defect prediction (SDF) is considered a crucial software quality assurance technique, that 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 (FSS) 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 multifilter 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 algorithm to a binary version where eight different transfer functions from two groups, to examine the probabilities of updating the process of choosing features from a binary vector, S-shaped and V-shaped models are adopted. 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 experiment 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 behaviour 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 $n_c$ represents the number of Whales, and $x_i^t$ represents the value for the position of the (*ith*) 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*):

$$whales(X) = \begin{bmatrix} x_1^1 & x_2^1 & \dots & x_d^1 \\ x_1^2 & x_2^2 & \dots & x_d^2 \\ \vdots & \vdots & \dots & \vdots \\ x_1^n & x_2^n & \dots & x_d^n \end{bmatrix}$$

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:

$$D = |C.X^*(t) - X\,(t)\,| \tag{1}$$

$$X(t+1) = X^*(t+1) - A.D \tag{2}$$

where *t* 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:

$$A = 2ar - a \tag{3}$$

$$C = 2r \tag{4}$$

The variable "*r*" 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:

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 [-*a*, *a*].

Spiral updating position: this process includes calculating the distance between the whale and the prey. Then, the spiral equation is utilized to imitate the movement resembling a helix, as shown in equation 5.

$$X(t+1) = D^1 e^{bl} \cos(2\pi l) + X^*(t) \tag{5}$$

where $l$ represents a random number within [−1, 1] and $b$ is a constant. It is assumed that there is a 50% chance of selecting either the shrinking encircling mechanism or the spiral model. It can be represented mathematically as shown in equation 6:

$$X(t+1) = \begin{cases} X^*(t) - AD & if\ p < 0.5 \\ D^1 e^{bl} \cos(2\pi l) + X^*(t) & if\ p \geq 0.5 \end{cases} \tag{6}$$

where $p$ is a uniformly distributed random number. In the exploration phase, to encourage the agent to move away from its current position, random values within the range of 1 < $A$ < -1 are utilized, as shown in equations 7 and 8:

$$\mathrm{D} = |\mathrm{CX}_{\mathrm{rand}} - \mathrm{X}| \tag{7}$$

$$\mathrm{X}(\mathrm{t}+1) = \mathrm{X}_{\mathrm{rand}} - \mathrm{AD} \tag{8}$$

In the binary WOA (BWOA) [19], whales move inside a binary search space instead of a continuous search space in order to modify their positions. So, to solve the feature selection problem, the solutions must be represented as 0 and 1 only. Actually, there are two versions of BWOA which are S-BWOA and V-BWO based on the used transfer functions, either the S-shaped function or the V-shaped function [14].

## 2. 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):

$$T1(x) = \frac{\sqrt{|x|}}{\sqrt{|A|}} \tag{9}$$

$$T2(x) = \frac{|x|}{|A|} \tag{10}$$

$$T3(x) = \frac{\sqrt[3]{|x|}}{\sqrt[3]{A}} \tag{11}$$

$$T4(x) = \frac{\sqrt[4]{|x|}}{\sqrt[4]{|A|}} \tag{12}$$

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

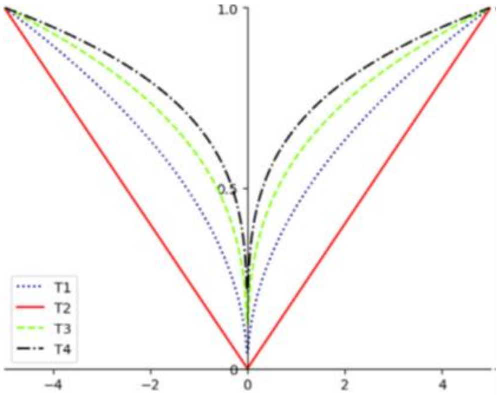

*Fig. 1. The curves of Taper-shaped transfer functions [12]*

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

$$x_b = \begin{cases} 1, & if\ T_k(x) \leq T_s \\ 0, & otherwise \end{cases} \tag{13}$$

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

$$f\left(x_{i,t}\right) = \propto C_r\left(x_{i,t}\right) + \beta \frac{|X|}{|K|} \tag{14}$$

where $C_r$ represents the error rate of the classification, $X$ represents the number of selected features by whale $(x_{i,t})$, $K$ represents all the features, $\propto$ =0.99 and $\beta$ =1-$\propto$. Algorithm 1 shows the steps of the proposed T-BWOA.

```
Algorithm 1: (T-BWOA-KNN)
start
The inputs: a number of whales (N) and
   maximum number of iterations (t).
The output: the best whale's positions
set the Initial values of a and N
compute the fitness value for the whales according
 to equation (4) and find the best search agent (X^*)
while stop condition( maximum iteration (t)) has not satisfied do
    for i = 1: N
            Calculate and Update the following parameters:
                a = 2 - t * (2 / t)
                A = 2 * a * rand() – a
                C= 2 * rand()
                P= rand(),
                L= random number in [-1,1]
        if  p >= 0.5 then
            $X(t+1) = D' * e^{b*l} * \cos(2\pi l) + x^*(t)$
                $D' = x^*(t) - x(t)$
        Else (p<0.5) do
                if (|A| < 1)then
                $x(t+1) = x^*(t) - A * D$
                $D = |C * x^*(t) - x(t)|$
                Else (|A| ≥ 1)
                $x(t+1) = x_{rand} - A * D_x$
                $D_x = |C * x_{rand} - x|$
                End if
          End if
    Convert  $x(t+1)$ to binary space using four types of taper
    shaped transfer functions  based on equation (13) for each t value
    End for
    calculate the fitness value using KNN classifier for each whale
    based on equation (14)
    Update the value of x^* , if there is a better solution
    t=t+1
end while
```

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

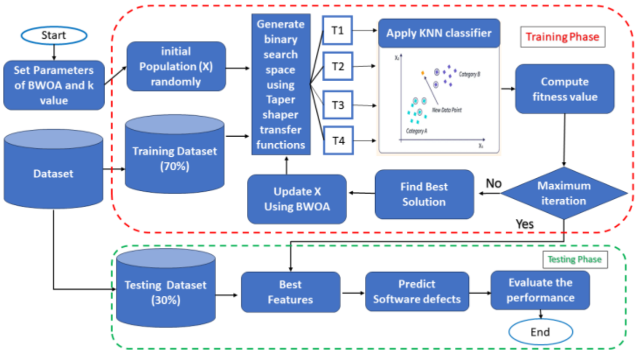

*Fig. 2. The framework of the proposed T-BWOA-KNN*

## 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*

| Dataset | Attribute (features) | No. of records | Defective records | Non-defective records |
|---|---|---|---|---|
| CM1 | 37 | 327 | 42 | 285 |
| JM1 | 21 | 9593 | 1759 | 7834 |
| KC1 | 21 | 2096 | 325 | 1771 |
| KC3 | 39 | 194 | 36 | 158 |
| MW1 | 37 | 250 | 25 | 225 |
| PC1 | 37 | 679 | 55 | 624 |
| PC2 | 36 | 1585 | 16 | 1569 |
| PC3 | 37 | 1125 | 140 | 985 |
| PC4 | 37 | 1270 | 176 | 1094 |
| PC5 | 39 | 17186 | 516 | 16670 |
| Tomcat | 20 | 858 | 77 | 781 |

## 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).

$$\text{Normalized value } (x') = (x-\min)/(\max-\min) \qquad (15)$$

## 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*

| Predicted output | Actual output | |
|---|---|---|
| | Positive | Negative |
| Positive | TP | FP |
| Negative | FN | TN |

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 carve (AUC), sensitivity (SN), specificity (SP), number of selected features (SF), error rate (ER), and G-mean.

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

$$ACC = (TP + TN) / (TP + TN + FP + FN) \qquad (16)$$

Also, area under carve (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)

$$AUC=(1+TPR-FPR)/2 \qquad (17)$$

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

$$TPR=TP/ (TP+ FN) \qquad (18)$$

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).

$$FPR=FP/(FP+TN) \qquad (19)$$

The false positive rate is typically expressed as a percentage or a decimal value between 0 and 1. A lower FPR indicates a more accurate model, as it means fewer negative instances are being misclassified as positive.

In addition, G-mean is used to show the efficiency of both sensitivity (SN) and specificity (SP) together [10], which is calculated using equation (20).

$$\text{G-mean}= \sqrt{} (SN*SP) \qquad (20)$$

where SN is the probability of correct classification for the positive instances. SP represents the probability of correct classification for the negative instances, which are calculated based on equations (21) and (22), respectively.

$$SN=TP /(TP +FN) \qquad (21)$$

$$SP= TN/(TN +FP) \qquad (22)$$

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

$$ER=1- \text{Accuracy} \qquad (23)$$

## 3.4. Parameters configuration

The values of the parameters for each experiment are set as shown in table 3.

*Table 3. The values of parameters for each experiment*

| Parameter's name | Parameters values |
|---|---|
| Number of whales | 120 |
| Number of iterations | 100 |
| Number of dimensions | Number of total features for each dataset |
| Initial values | [0,1] |
| *b* | 1 |
| *K* | 5 |

## 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)
$$R_i = \sum_{j=}^{m} r_{i,j} \qquad (24)$$
2. The average value of (R) is calculated using equation (25)
$$\bar{R} = \frac{1}{n}\sum_{i=1}^{n} R_i \qquad (25)$$
3. The sum of squared deviations, S, is calculated using equation (26)
$$S = \sum_{i=1}^{n}(R_i - \bar{R})^2 \qquad (26)$$
4. The Kendall's W coefficient is calculated using equation (27)
$$W = \frac{12*S}{m^2(n^3-n)} \qquad (27)$$

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

## 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 SDP problem in terms of accuracy rate (ACC), the average number of selected features (SF), sensitivity (SN), specificity (SP), area under carve (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 whose 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-BWO, 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 G-mean, as shown in table 7, although T1-BWOA, T3-BWOA, and T4-BWOA have produced the highest G-mean for three different datasets which are shaded by gray colour. The T4-BWOA consider the best in terms of (G-mean) mean (0.456%).

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).

*Table 4. The obtained results of T-BWOA in terms of accuracy (%)*

| Dataset | T1-BWOA | T2-BWOA | T3-BWOA | T4-BWOA |
|---|---|---|---|---|
| Cm1 | 0.851 | 0.849 | 0.84 | 0.841 |
| JM1 | 0.958 | 0.958 | 0.956 | 0.958 |
| KC1 | 0.846 | 0.850 | 0.842 | 0.842 |
| KC3 | 0.790 | 0.781 | 0.786 | 0.798 |
| MW1 | 0.896 | 0.86 | 0.900 | 0.888 |
| PC1 | 0.913 | 0.902 | 0.901 | 0.884 |
| PC2 | 0.988 | 0.990 | 0.989 | 0.988 |
| PC3 | 0.853 | 0.868 | 0.852 | 0.861 |
| PC4 | 0.862 | 0.843 | 0.869 | 0.867 |
| PC5 | 0.973 | 0.974 | 0.976 | 0.977 |
| tomcat | 0.879 | 0.892 | 0.900 | 0.89 |
| Mean | 0.891 | 0.887 | 0.891 | 0.890 |
| Total shaded | 3 | 4 | 3 | 3 |

*Table 5. The obtained results of T-BWOA in terms of sensitivity (%)*

| Dataset | T1-BWOA | T2-BWOA | T3-BWOA | T4-BWOA |
|---|---|---|---|---|
| Cm1 | 0.084 | 0.038 | 0.031 | 0.046 |
| JM1 | 0.866 | 0.862 | 0.861 | 0.858 |
| KC1 | 0.368 | 0.366 | 0.400 | 0.347 |
| KC3 | 0.127 | 0.136 | 0.127 | 0.164 |
| MW1 | 0.238 | 0.175 | 0.263 | 0.2 |
| PC1 | 0.194 | 0.131 | 0.194 | 0.238 |
| PC2 | 0.02 | 0.100 | 0.02 | 0.100 |
| PC3 | 0.260 | 0.190 | 0.183 | 0.214 |
| PC4 | 0.283 | 0.038 | 0.315 | 0.262 |
| PC5 | 0.429 | 0.862 | 0.479 | 0.512 |
| tomcat | 0.178 | 0.366 | 0.161 | 0.226 |
| Mean | 0.277 | 0.296 | 0.275 | 0.287 |
| Total shaded | 4 | 3 | 4 | 2 |

*Table 6. The obtained results of T-BWOA in terms of specificity (%)*

| Dataset | T1-BWOA | T2-BWOA | T3-BWOA | T4-BWOA |
|---|---|---|---|---|
| Cm1 | 0.968 | 0.973 | 0.964 | 0.962 |
| JM1 | 0.978 | 0.980 | 0.978 | 0.980 |
| KC1 | 0.934 | 0.940 | 0.923 | 0.933 |
| KC3 | 0.945 | 0.932 | 0.940 | 0.947 |
| MW1 | 0.975 | 0.942 | 0.976 | 0.970 |
| PC1 | 0.974 | 0.968 | 0.962 | 0.939 |
| PC2 | 0.999 | 1 | 1 | 0.998 |
| PC3 | 0.938 | 0.964 | 0.947 | 0.953 |
| PC4 | 0.955 | 0.946 | 0.958 | 0.964 |
| PC5 | 0.990 | 0.991 | 0.992 | 0.991 |
| tomcat | 0.948 | 0.970 | 0.972 | 0.955 |
| Mean | 0.964 | 0.964 | 0.964 | 0.962 |
| Total shaded | 1 | 5 | 4 | 3 |

*Table 7. The obtained results of T-BWOA in terms of G-mean(%)*

| Dataset | T1-BWOA | T2-BWOA | T3-BWOA | T4-BWOA |
|---|---|---|---|---|
| Cm1 | 0.210 | 0.101 | 0.091 | 0.128 |
| JM1 | 0.920 | 0.919 | 0.918 | 0.917 |
| KC1 | 0.579 | 0.580 | 0.604 | 0.561 |
| KC3 | 0.300 | 0.318 | 0.259 | 0.316 |
| MW1 | 0.417 | 0.396 | 0.465 | 0.434 |
| PC1 | 0.335 | 0.301 | 0.360 | 0.398 |
| PC2 | 0.045 | 0.197 | 0.045 | 0.171 |
| PC3 | 0.484 | 0.420 | 0.408 | 0.444 |
| PC4 | 0.494 | 0.419 | 0.547 | 0.493 |
| PC5 | 0.647 | 0.629 | 0.689 | 0.710 |
| tomcat | 0.354 | 0.255 | 0.354 | 0.451 |
| Mean | 0.435 | 0.412 | 0.430 | 0.456 |
| Total shaded | 3 | 2 | 3 | 3 |

*Table 8. The obtained results of T-BWOA in terms of AUC (%)*

| Dataset | T1-BWOA | T2-BWOA | T3-BWOA | T4-BWOA |
|---|---|---|---|---|
| Cm1 | 0.526 | 0.506 | 0.497 | 0.504 |
| JM1 | 0.922 | 0.921 | 0.920 | 0.919 |
| KC1 | 0.651 | 0.653 | 0.662 | 0.640 |
| KC3 | 0.536 | 0.534 | 0.534 | 0.555 |
| MW1 | 0.606 | 0.558 | 0.619 | 0.585 |
| PC1 | 0.584 | 0.549 | 0.578 | 0.588 |
| PC2 | 0.509 | 0.550 | 0.51 | 0.549 |
| PC3 | 0.599 | 0.577 | 0.565 | 0.583 |
| PC4 | 0.619 | 0.575 | 0.637 | 0.613 |
| PC5 | 0.710 | 0.699 | 0.736 | 0.751 |
| tomcat | 0.563 | 0.533 | 0.566 | 0.590 |
| Mean | 0.620 | 0.605 | 0.620 | 0.625 |
| Total shaded | 3 | 1 | 3 | 4 |

*Table 9. The obtained results of T-BWOA in terms of error rate (ER)(%)*

| Dataset | T1-BWOA | T2-BWOA | T3-BWOA | T4-BWOA |
|---|---|---|---|---|
| Cm1 | 0.148 | 0.151 | 0.16 | 0.159 |
| JM1 | 0.042 | 0.042 | 0.044 | 0.042 |
| KC1 | 0.154 | 0.150 | 0.158 | 0.158 |
| KC3 | 0.210 | 0.219 | 0.214 | 0.202 |
| MW1 | 0.104 | 0.140 | 0.100 | 0.112 |
| PC1 | 0.087 | 0.098 | 0.099 | 0.116 |
| PC2 | 0.012 | 0.010 | 0.011 | 0.012 |
| PC3 | 0.147 | 0.132 | 0.148 | 0.139 |
| PC4 | 0.138 | 0.157 | 0.131 | 0.133 |
| PC5 | 0.027 | 0.026 | 0.024 | 0.023 |
| tomcat | 0.121 | 0.108 | 0.100 | 0.11 |
| Mean | 0.1081 | 0.1120 | 0.1080 | 0.1096 |
| Total shaded | 3 | 4 | 3 | 3 |

*Table 10. The averaged selected features obtained by the proposed method (T-BWOA-KNN)*

| Datasets | T1-BWOA-KNN | T2-BWOA-KNN | T3-BWOA-KNN | T4-BWOA-KNN |
|---|---|---|---|---|
| CM1 | 3 | 2 | 4 | 4 |
| JM1 | 2 | 2 | 2 | 2 |
| KC1 | 7 | 4 | 7 | 6 |
| KC3 | 4 | 4 | 6 | 4 |
| MW1 | 5 | 6 | 5 | 6 |
| PC1 | 5 | 3 | 4 | 4 |
| PC2 | 2 | 2 | 3 | 2 |
| PC3 | 5 | 2 | 7 | 6 |
| PC4 | 8 | 5 | 7 | 7 |
| PC5 | 3 | 2 | 4 | 3 |
| TOMCAT | 5 | 3 | 5 | 5 |
| Total shaded | 4 | 9 | 2 | 3 |

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-KN, 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 KC3 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.

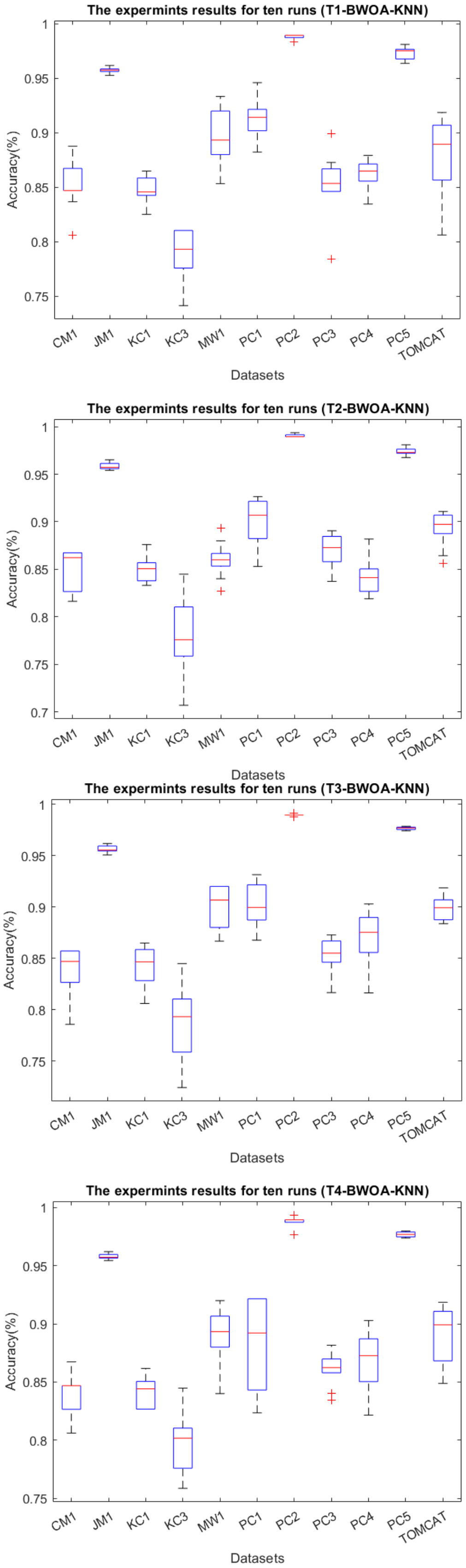

*Fig. 3. Boxplot of all datasets for the proposed method(T1~T4-BWOA-KNN) in terms of accuracy*

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 selected features and 21% 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*

| Dataset | The Proposed method | [5] | | [13] | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | T2-BWOA-KNN | NB+EWFS | DT+EWFS | CFS+KNN+GA | CFS+KNN+BAT | CFS+KNN+PSO | CFS+KNN+FS | CFS+KNN+AS | CNS+KNN+GA | CNS+KNN+BAT | CNS+KNN+PSO | CNS+KNN+FS | CNS+KNN+AS |
| CM1 | 2 | 4 | 7 | 7 | 5 | 8 | 7 | 5 | 12 | 12 | 6 | 15 | 8 |
| JM1 | 2 | / | / | / | / | / | / | / | / | / | / | / | / |
| KC1 | 4 | 2 | 4 | 8 | 4 | 8 | 4 | 2 | 11 | 17 | 16 | 16 | 17 |
| KC3 | 4 | 3 | 3 | 2 | 2 | 3 | 3 | 2 | 9 | 17 | 6 | 12 | 13 |
| MW1 | 6 | 3 | 3 | 8 | 9 | 7 | 9 | 7 | 11 | 17 | 8 | 17 | 13 |
| PC1 | 3 | 5 | 6 | / | / | / | / | / | / | / | / | / | / |
| PC2 | 2 | / | / | 5 | 5 | 5 | 5 | 6 | 15 | 17 | 9 | 17 | 16 |
| PC3 | 2 | 3 | 5 | / | / | / | / | / | / | / | / | / | / |
| PC4 | 5 | 6 | 3 | / | / | / | / | / | / | / | / | / | / |
| PC5 | 2 | 3 | 6 | / | / | / | / | / | / | / | / | / | / |
| TOMCAT | 3 | 6 | 4 | / | / | / | / | / | / | / | / | / | / |

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

| Dataset | proposed | [5] | | [13] | | | | | [29] | | [14] |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | T2+BWOA+KNN | NB+EWFS | DT+EWFS | CFS+KNN+GA | CFS+KNN_+BAT | CFS+KNN_+PSO | CFS+KNN_+FS | CFS+KNN+AS | NB+RMFFS | DT+RMFFS | MLP+FS |
| CM1 | 84.9 | 87.16 | 86.63 | 77.68 | 77.37 | 80.43 | 81.04 | 78.59 | 73.3 | 61.8 | 89.795 |
| JM1 | 95.8 | / | / | / | / | / | / | / | / | / | 80.44 |
| KC1 | 85.0 | 75.3 | 75.9 | 71.26 | 72.98 | 70.57 | 70.40 | 71.69 | 78.2 | 65.1 | 77.6504 |
| KC3 | 78.1 | 82.47 | 85.41 | 78.87 | 78.87 | 75.77 | 74.23 | 75.26 | 71.0 | 68.1 | 82.758 |
| MW1 | 86.0 | 90.0 | 89.2 | 84 | 84.4 | 84 | 84 | 82 | / | / | 92.000 |
| PC1 | 90.2 | 91.9 | 91.9 | / | / | / | / | / | / | / | 96.078 |
| PC2 | 99.0 | / | / | 96.54 | 96.12 | 96.81 | 95.84 | 95.29 | / | / | 97.695 |
| PC3 | 86.8 | 84.59 | 86.54 | / | / | / | / | / | 79.8 | 66.6 | 85.126 |
| PC4 | 84.3 | 82.67 | 88.89 | / | / | / | / | / | / | / | 88.97 |
| PC5 | 97.4 | 74.87 | 76.04 | / | / | / | / | / | / | / | 74.803 |
| TOMCAT | 89.2 | 90.96 | 92.02 | / | / | / | / | / | / | / | / |

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

| Mutual Datasets | W | P | Rank methods | |
|---|---|---|---|---|
| CM1,<br>KC1,<br>KC3,<br>MW1 | 0.745 | 0.002 | T1_BWOA_KNN | 8.25 |
| | | | T3_BWOA_KNN | 7.25 |
| | | | T4_BWOA_KNN | 7.25 |
| | | | T2_BWOA_KNN | 6.75 |
| | | | CFS_KNN_BAT | 4.12 |
| | | | CFS_KNN_GA | 3.38 |
| | | | CFS_KNN_PSO | 3 |
| | | | CFS_KNN_FS | 2.5 |
| | | | CFS_KNN_AS | 2.5 |

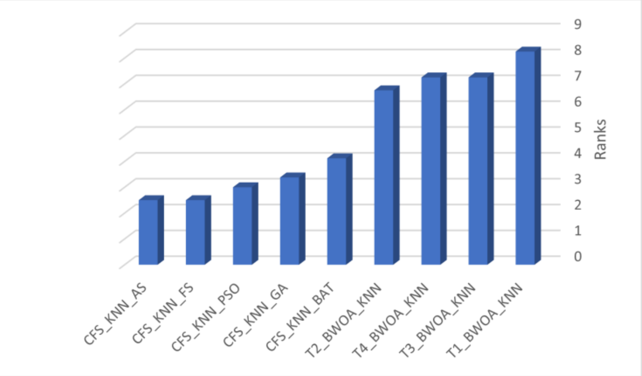

*Fig. 4. The ranking of (T-BWOA-KNN) with other methods from the literature*

## 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

[1] Adamu A. et al.: An hybrid particle swarm optimization with crow search algorithm for feature selection. Machine Learning with Applications 6, 2021, 100108.

[2] Al Qasem O., Akour M.: Software fault prediction using deep learning algorithms. International Journal of Open Source Software and Processes (IJOSSP) 10(4), 2019, 1–19.

[3] Balogun A. O. et al.: Performance analysis of feature selection methods in software defect prediction: a search method approach. Applied Sciences 9(13), 2019, 2764.

[4] Balogun A. O. et al.: Rank aggregation-based multi-filter feature selection method for software defect prediction. Advances in Cyber Security: Second International Conference – ACeS 2020, 2021.

[5] Balogun A. O. et al.: Software defect prediction using wrapper feature selection based on dynamic re-ranking strategy. Symmetry 13(11), 2021, 2166.

[6] Balogun A. O. et al.: An adaptive rank aggregation-based ensemble multi-filter feature selection method in software defect prediction. Entropy 23(10), 2021, 1274.

[7] De Souza R. C. T. et al.: A V-shaped binary crow search algorithm for feature selection. IEEE Congress on Evolutionary Computation – CEC, 2018.

[8] Fan G. et al.: Software defect prediction via attention-based recurrent neural network. Scientific Programming 2019, 6230953.

[9] Gad A. G. et al.: An improved binary sparrow search algorithm for feature selection in data classification. Neural Computing and Applications 34(18), 2022, 15705–15752.

[10] Hamed A. et al.: Algamal, Improving binary crow search algorithm for feature selection. Journal of Intelligent Systems 32(1), 2023, 20220228.

[11] Hassouneh Y. et al.: Boosted whale optimization algorithm with natural selection operators for software fault prediction. IEEE Access 9, 2021, 14239–14258.

[12] He Y. et al.: Novel binary differential evolution algorithm based on Taper-shaped transfer functions for binary optimization problems. Swarm and Evolutionary Computation 69, 2022, 101022.

[13] Hossin M., Sulaiman M. N.: A review on evaluation metrics for data classify-cation evaluations. International journal of data mining & knowledge management process 5(2), 2015.

[14] Hussien A. G. et al.: Binary whale optimization algorithm for dimensionality reduction. Mathematics 8(10), 2020, 1821.

[15] Hussien A. G. et al.: S-shaped binary whale optimization algorithm for feature selection. Recent Trends in Signal and Image Processing: ISSIP 2017.

[16] Iqbal A., Aftab S.: A Classification Framework for Software Defect Prediction Using Multi-filter Feature Selection Technique and MLP. International Journal of Modern Education & Computer Science 12(1), 2020.

[17] Jureczko M., Madeyski L.: Towards identifying software project clusters with regard to defect prediction. Proceedings of the 6th international conference on predictive models in software engineering. 2010.

[18] Landis J. R., Koch G. G.: The measurement of observer agreement for categorical data biometrics 1977, 159–174.

[19] Legendre P.: Species associations: the Kendall coefficient of concordance revisited. Journal of agricultural, biological, and environmental statistics 10, 2005, 226–245.

[20] Mirjalili S., Lewis A.: The whale optimization algorithm. Advances in engineering software 95, 2016, 51–67.

[21] Rathore S. S., Kumar S.: A decision tree logic based recommendation system to select software fault prediction techniques. Computing 99, 2017, 255–285.

[22] Shepperd M. et al.: Data quality: Some comments on the nasa software defect datasets. IEEE Transactions on software engineering 39(9), 2013, 1208–1215.

[23] Shepperd M. et al.: Nasa mdp software defects data sets. Figshare Collection, 2018.

[24] Tohka J., Van Gils M.: Evaluation of machine learning algorithms for health and wellness applications: A tutorial. Computers in Biology and Medicine 132, 2021, 104324.

[25] Tumar I. et al.: Enhanced binary moth flame optimization as a feature selection algorithm to predict software fault prediction. IEEE Access 8, 2020, 8041–8055.

[26] Turabieh H., Mafarja M., Li X.: Iterated feature selection algorithms with layered recurrent neural network for software fault prediction. Expert systems with applications 122, 2019, 27–42.

[27] Witten I. H., Frank E., Hall M. A.: Data Mining: Practical Machine Learning Tools and Techniques. Morgan Kaufmann Publishers Inc. 2011.

[28] Xu Z. et al.: Software defect prediction based on kernel PCA and weighted extreme learning machine. Information and Software Technology 106, 2019, 182–200.

[29] Zhou T. et al.: Improving defect prediction with deep forest. Information and Software Technology 114, 2019, 204–216.

[30] Zhu K. et al.: Software defect prediction based on enhanced metaheuristic feature selection optimization and a hybrid deep neural network. Journal of Systems and Software 180, 2021, 111026.

**M.Sc. Zakaria A. Hamed Alnaish**
e-mail: zakriahamoalnaish@uomosul.edu.iq

He is master in computer science from university technology Malaysia, faculty of computing and got best student award. Currently, he works as lecturer in College of Science/University of Mosul and Ph.D. candidate in Computer Science Department, College of Computer Science and Mathematics, University of Mosul.
Research interest: machine learning, fuzzy logic, optimization algorithms, software engineering.

http://orcid.org/0000-0002-7597-5326

**Prof. Dr. Safwan O. Hasoon**
e-mail: Dr.safwan@uomosul.edu.iq

He is a doctor and full professor in artificial intelligence. He works College of Computer Science and Mathematics, University of Mosul.
Research interest: machine learning, artificial intelligence, artificial neural network.

http://orcid.org/0000-0002-3653-3568