Research Article

Breast cancer diagnosis using a multi-verse optimizer-based gradient boosting decision tree

Hamed Tabrizchi[1] · Mohammad Tabrizchi[2] · Hamid Tabrizchi[3]

Received: 13 February 2020 / Accepted: 19 March 2020 / Published online: 24 March 2020

Abstract

Breast cancer is among the most common cancers women got, which can be effectively cured providing that it is diagnosed at the early stages. In the current study, we attempted to classify breast cancer into two groups of malignant and benign by proposing a new ensemble learning method using Multi-Verse Optimizer (MVO) and Gradient Boosting Decision Tree (GBDT). Moreover, the prediction rate of GBDT has been shown to be desirable, its efficiency and classification accuracy are significantly dependent on feature selection and parameter setting. Based on the MVO, we attempted to propose an efficient approach to optimize feature selection and GBDT's parameters at the same time. In other words, the MVO algorithm is able to play the role of a tuner to set the GBDT's main parameters and optimize feature selection results. To implement and test the proposed approach, standard criteria (i.e. accuracy, sensitivity, specificity, etc.) was used for performance evaluation. Also, the datasets of Wisconsin Diagnostic Breast Cancer and Wisconsin Breast Cancer were considered for this purpose. Comparing the results of GBDT–MVO model with other proposed models demonstrated that this model is more precise and has considerably lower variance in the case of a breast cancer diagnosis.

Keywords Cancer diagnoses · Classification · Gradient boosting decision tree · Multiverse optimizer

1 Introduction

As we all know, with a significant mortality rate, breast cancer has been among the most prevalent cancers in recent decades. Early detection of breast cancer greatly increases the likelihood of patient survival. So, this cancer needs a precise and reliable approach to be diagnosed on time. Fortunately, in the last decade, the outcomes of breast cancer have been improved because of the efficient diagnosis approaches and enhanced treatment methods. As mentioned, the most important objective is to diagnose this cancer earlier and more accurately. In this regard, many approaches (such as screening) have been proposed to detect different kinds of cancers before symptom appearance and predict treatment outcomes. However, there are many medical datasets that can be employed in the field of cancer studies. In this regard, the precise prediction of disease outcome has been very interesting and challenging. It is obvious that the decisions made by physicians based on data evaluation can be considered the most effective factor for diagnosis. Accordingly, ML methods can significantly help researchers in this field. In fact, the relationships and patterns within the datasets can be discovered by these methods to predict the outcomes of disease [1]. In addition, the classification methods proposed based on rules and ML techniques are able to minimize weak decisions made by inexperienced or exhausted experts and facilitate the accurate and prompt data analysis. In other words, the incorrect decisions made by physicians can be significantly reduced using ML models. These models employ the datasets collected using historical cases to find

✉ Hamed Tabrizchi, hamed.tabrizchi@math.uk.ac.ir | [1]Department of Computer Science, Shahid Bahonar University of Kerman, Kerman, Iran. [2]Department of Medicine, Eastern Mediterranean University, Famagusta, North Cyprus, Cyprus. [3]Department of Pathology, Afzalipoor School of Medicine, Kerman, Iran.

SN Applied Sciences (2020) 2:752 | https://doi.org/10.1007/s42452-020-2575-9

the relationships and patterns among the various cases and forecast the related outcomes. However, as a supervised ML method, classification algorithms can learn how to classify new observations based on the given input data. Up to this moment, the world of machine learning has presented a lot of methods with specific advantages and disadvantages that have the potential to be used in a variety of applications. One of the sophisticated ensemble models used in different classification and regression problems is GBDT. It is a vivid fact that this famous machine learning algorithm leading to a great deal of successful achievement across many domains. In fact, this method increases the accuracy of prediction models by taking advantage of an ensemble of weak prediction models. Compared to other ensemble techniques, the use of the GBDT provides significant advantages such as high speed and high accuracy. In other words, GBDT builds weak models with affordable computation costs and these weak models allow the algorithm to learn slowly to make adjustments in new areas where it does not perform well. However, it is undeniable that the weak models witnessed a high error rate and the boosting methods aim to build a sequential model in order to reduce the errors. One of the important problems that all machine learning algorithms encounter is parameter tuning which can be defined as the problem of choosing a set of optimal parameters to control the learning process. In addition, the performance of the learning model is significantly influenced by the values set for these parameters. Due to the fact that GBDT provides high predictive accuracy and great deals of flexibility regarding using different loss functions, overfitting is able to occur in GBDT because the process of parameter tuning regarding many parameters that requires a large grid search during tuning in order to manage the performance of the GBDT. In general, GBDT needs two types of parameters to be tuned as a tree, based on boosting parameters. To solve the parameter setting problem, this study aims to solve two critical problems. The first is to optimize two boosting parameters namely the learning rate and the number of the sequential trees (estimators) [2]. The second is to optimize trees based on the following parameters; the maximum depth of a tree (max depth) and the minimum number of the samples of terminal nodes (min leaf nodes). When taking advantage of the GDBT algorithm to train a learning model, the correct parameter setting (which can be adjusted depending on a dataset) is significantly critical. Although GBDT is highly robust against the over-fit issue caused by increased number of estimators (trees), a high learning rate can lead the GBDT model to overfit. Also, it is worth noting that reduced learning rate and increased number of trees increase computation complexity. However, the maximum admissible interaction level among variables is controlled by the number of terminal nodes which itself is limited by maximum depth [3]. In other words, the deeper tree, the more the splits and captured information. It is worth noting that large depth values may lead to over-fitting models that are able to predict all training data but not able to generalize results for new data. Another important challenge that all of the learning algorithms encounter is to select the best representative subset of features among all possible representative subsets (2^n) that can be used in the training process. In the current study, an efficient learning method is proposed based on the MVO algorithm to optimize the feature selection process and the parameters of GBDT simultaneously. In other words, the MVO algorithm plays the role of a tuner to find the best values for GBDT's parameters and the optimal set of features to maximize GBDT's accuracy. In fact, in this study, we use MVO to optimize GBDT for the first time. The proposed approach is implemented and tested based on standard criteria (i.e. sensitivity, specificity, F-measure, etc.) to classify date available in two well-known cancer datasets called Wisconsin Breast Cancer and Wisconsin Diagnostic Breast Cancer.

The main contributions of this work can be summarized as follows:

- This work aims to present novel effective diagnostic techniques based on a popular supervised learning method called GBDT.
- In order to improve the performance of the GBDT, this research focuses on two important factors (namely feature selection and parameter tuning) that are optimized using the multi-verse optimizer.
- The main purpose of this study is to increase classification accuracy and prevent overfitting issues; In order to achieve this goal, K-fold cross-validation is used during the optimization of the GBDT.

The rest of this paper is organized as follows. Section 2 reviews and summarizes previous studies in the field of breast cancer detection and ensemble learning methods. GBDT and MVO algorithm are presented in Sect. 3. Also, Sect. 4 presents and discusses our proposed approach to optimize feature selection process and the values of GBDT's parameters. Section 5 analyzes and discusses the experimental results. Finally, the paper comes to a conclusion in Sect. 6.

2 Related work

So far, many Machine learning and soft computing approaches have been applied to breast cancer diagnosis problems due to their cost-effectiveness and high accuracy. The most important approaches in this filed are as follows; support vector machines (SVMs) [4–6], Decision trees [7–9], Artificial neural network (ANN) [10–14], Naive Bayes classifier [15], K-nearest neighbour [16], and ensemble methods [17–20]. It is undeniable that majority of the mentioned learning approaches have to deal with difficult challenges such as feature subset selection, along with the parameter tuning in their training procedure. Because the accuracy of their classification results depends largely on both. For this reason, metaheuristics are used in order to deal with mentioned drawbacks by the process of searching optimal solutions. In fact, the process of searching is able to take multiple agents by using a combination of rules or mathematical equations during several iterations. In addition, most of the rules or mathematical equations used by most of the metaheuristic methods have been inspired by the living and survival systems of insects, animals, and birds. Due to the noticeable success of metaheuristic algorithms in solving a lot of optimization problems in a wide range of applications, there are various types of metaheuristic algorithms include Genetic algorithm [21, 22], Firefly Algorithm [23], Particle swarm optimization [24, 25], Ant Colony Optimization [26], Bat algorithm [27], Whale Optimization Algorithm [28], Artificial fish swarm [29], and Grey wolf optimizer [30] has been extensively reported in recent literature. To classify breast tumors into cancerous and non-cancerous ones, an ensemble learning method was proposed by Vinod Jagannath Kadam et al. [17] based on SoftMax Regression and Sparse Autoencoders. The results of their study demonstrated its efficiency for breast tumor classification. In fact, this ensemble approach outperformed many ML and soft computing classifiers including KNN, SVM, Decision Tree, etc. Nilashi et al. [31] presented a novel knowledge-based system for breast cancer classification using fuzzy logic method. The goal of this research was to diagnose breast cancer disease using clustering, noise removal, and classification techniques. In this approach, the data were clustered in similar groups using the Expectation–Maximization (EM) and fuzzy roles were produced using Regression Trees to classify breast cancer disease in the knowledge-based fuzzy system. By taking advantage of Wisconsin Diagnostic Breast Cancer dataset, the authors demonstrated that the proposed knowledge-based system is able to enhance the prediction accuracy considerably. In fact, one of the challenging objectives in most of the related studies is to set model's parameters in an optimized way. In some studies, the meta-heuristic algorithms were combined with ML models to better tune the model's parameters. In this regard, a swarm intelligence technique was combined with an SVM classifier by Chen et al. [32] to diagnose breast cancer. The focus of this work was on feature selection and model selection based on the swarm optimization approach. The comparison between this model and the grid search method showed that this method is able to provide better model parameters, discriminative feature subset, and prediction accuracy using a smaller number of support vectors for training. Also, Chauhan et al. [33] took advantage of a differential evolution method to improve the wavelet neural network's training process by finding the best values for parameters. They tested this network on three standard datasets (including WBC dataset) and three bank bankruptcy datasets. The results of this work demonstrated that the proposed model is able to relatively high generalization. In another work, Jain et al. [34] integrated correlation-based feature selection with Binary Particle Swarm Optimization to propose a hybrid model to classify cancers. In this model, the biological samples of binary and multi-class cancers are classified using Naive–Bayes classifier by selecting a low-dimensional prognostic solution set. In this work, different datasets were used to evaluate the performance of the method. Accordingly, the results showed good classification accuracy. In a study conducted by Naveen et al. [35], the differential evolution method was combined with K-means to centralize data points and implement a radial basis function network that can be used as a supervised learning approach. By comparing their method with other existing ones (like threshold accepting trained wavelet neural network) on standard and bank bankruptcy datasets, they proved their method's good accuracy. In summary, most recent studies have focused on parameter tuning and feature selection objectives because the performance of a learning algorithm can be significantly influenced by these two important factors. There are many hybrid models proposed to configure these parameters systematically. However, it is worth noting that to prevent over-fitting issue, one should take into account suitable evaluation measures for parameter tuning process. It is obvious that most of the proposed models require accurate statistical analyses to obtain desirable results when facing real data. Evaluating learning models based on only a percentage of the data causes a high risk for vital applications such as cancer diagnosis. In this regard, K-fold Cross-Validation can be used to divide data into multiple folds so that every single fold should be employed as testing set at some point. In this study, the K-fold Cross-Validation was used to evaluate the proposed approach precisely with various measures like accuracy, specificity, sensitivity, etc.

3 Methodology

In this section, the GBDT and MVO algorithms were explained in detail. Regarding the critical parameters of GBDT, the performance of each model was investigated using the MVO. This investigation aims to tune the parameters of the model and select vital features to improve the performance of the model. In addition, this investigation was performed based on the related performance evaluation methods like confusion matrix analysis.

3.1 Gradient boosting decision tree (GBDT)

The GBDT model was first introduced by Friedman [36] as a robust ensemble model. In fact, this method turns weak basic classifiers into strong ones by combining them. Unlike other similar techniques, the GBDT model uses function space for optimization purposes. Also, compared to linear models (including logistic regression), this model is more flexible, scalable, and robust against the complexities of non-linear problems.

According to Fig. 1, because of the hierarchical structure of non-linear decision boundaries, they can be naturally modeled using GBDT. In fact, the learning procedure of GBDT builds the base learners that can be maximally correlated with the loss function's negative gradient. Although traditional boosting methods use weighted positive and negative samples, the GBDT model follows the negative gradient's direction to converge globally. Totally, gradient boosting includes three parts; the loss function optimization, weak learner predictions, and loss function minimization by adding weak learners (Additive model). The loss function is defined based on the type of problem. In fact, in regression and classification problems, Mean Squared Error (MSE) and logarithmic loss are employed for this purpose. At each stage of boosting process, instead of starting from square one, only the unexplained loss from prior iterations should be optimized. Also, decision trees are employed as a weak learner and tress are added in single file to build an additive model which add weak learners for loss function minimization. Indeed, the trees existing in the model do not change. The loss accumulated during adding trees can be minimized using the gradient descent procedure [37–39]. Moreover, the dataset was shown by $\{x_i, y_i\}_{i=1}^{n}$ and loss function is SoftMax. The convergence of the model was guaranteed by taking advantage of the gradient descent algorithm. Also, in this model, M denotes the number of trees (the maximum number of iterations for training) and η denotes the learning rate defining the step size employed to combine the weights of individual trees in updates to intercept over-fitting issue. In addition, the minimum loss reduction needed for a further partition on a leaf node is shown by γ_m. The GBDT model functions as follows.

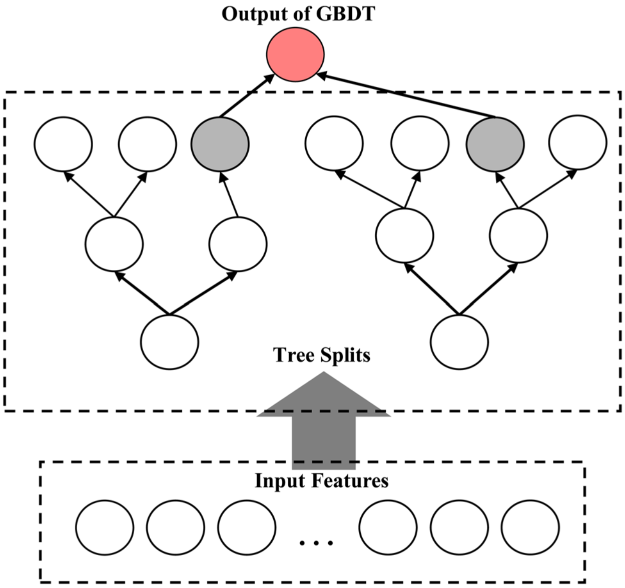


Fig. 1 The illustration of GBDT

Step 1 The initial constant value of the model γ is given

$$f_0(x) = argmin_\gamma \sum_{i=1}^{N} L(y_i, \gamma) \tag{1}$$

Step 2 determines the number of iterations; m = 1 to M

Step 2.1 based on Eq. (2), the minimum loss reduction and the step size used for combining the weights of individual trees can be calculated as follows:

$$\left(\gamma_m, \eta_m\right) = argmin_{\gamma,\eta} \sum_{i=1}^{N} L\left(y_i, f_{m-1}\left(x_{k,i}\right) + \eta b\left(x_{k,i};\gamma\right)\right) + \nu T + \frac{1}{2}\beta \parallel \gamma \parallel^2 \tag{2}$$

where T denotes the number of leaves on the tree. It should be noted that the loss function L() determines the model fitness with training data and measures the model complexity using the term $\eta b(x_{k,i};\gamma)$. Furthermore, the complexity of the model is penalized by the term $\nu T + \frac{1}{2}\beta \parallel \gamma \parallel^2$.

Step 2.2 updates the model as follows:

$$F_m(x) = F_{m-1}(x) + \eta_m b(x_{k,i};\gamma_m) \tag{3}$$

Step 3 returns $F_m(x)$ after using M additive functions to give the output.

As indicated in the following equation, given a sample X, GBDT uses M additive functions to give the output.

$$\hat{y}_{\text{GBDT}} = \sum_{i=1}^{M} \eta_m b(X;\gamma_m) \tag{4}$$

In fact, since a sequence of trees is computed in a GBDT model, the pseudo-residuals of the preceding trees are predicted by each successive tree, given an arbitrary differentiable loss function. The arbitrary loss function and the function calculating the corresponding negative gradient are defined by the user. Indeed, by aggregating the predictions, the loss function is minimized through training each added model. One of the most important factors to train a gradient boosting model is the number of trees because a too high number may cause overfitting and a too low number may cause underfitting.

3.2 Multi-verse optimizer (MVO)

The Multi-Verse optimizer is a metaheuristic algorithm that was first proposed in 2016 by Mirjalili et al. [40]. Many recent studies have used this algorithm in order to solve various problems in different applications. Due to the wide range of applications and their needs, other variants of MVO such as Binary Multi-verse optimizer [41] and Multi-Objective Multi-Verse Optimizer (MOMVO) [42] have been presented. This metaheuristic algorithm inspired by famous theory called Multi-Verse theory. The Multi-Verse theory was introduced based on three cosmologic concepts (black holes, white holes, and wormholes) and widely employed by physicists [43, 44]. Based on this theory, big bang occurred more than one time and each time a different universe was born with different physical laws. So, there are other universes in addition to the one we live in. According to the opinion of physicists, the primary part of the birth of a universe may be a white hole. However, black holes' behavior differs from white holes' so that everything even light beams can be attracted by their strong gravity. On the other hand, the different parts of a universe are connected together through wormholes. In fact, wormholes play the role of space travel tunnels in which objects can instantly travel within a universe. To model such a world, an inflation rate is assigned to each universe. Generally, the search space in population-based algorithms is divided into two phases namely, exploration and exploitation. In Multi-Verse Optimizer (MVO) [40], white holes and black holes perform the exploration phase. In addition, it is assumed that each solution is shown by a universe and each variable is an object in the universe. Also, the allocated inflation rate corresponds the fitness function value of the solution. As mentioned, each variable is an object in the universe that realizes the following rules.

1. If the inflation rate increases, the probability of having black holes decreases but the probability of having white holes increases.
2. In a universe with a high inflation rate, objects are sent through white holes but in a universe with low inflation rate the objects are received through black holes.
3. Regardless of the inflation rate, it is possible that wormholes move the objects randomly towards the best universe.

So, it can be concluded that it is very likely to move objects from a universe with a high inflation rate to another one with a low inflation rate. Accordingly, the average inflation rate is improved over iterations. An MVO method functions as follows.

Step 1 Initialize the universe (U), the maximum number of iterations (Max-iteration), the variable interval ([lb, ub]), and the universe position.

$$U = \begin{bmatrix} x_1^1 & \cdots & x_1^d \\ \vdots & \ddots & \vdots \\ x_n^1 & \cdots & x_n^d \end{bmatrix} \tag{5}$$

Step 2 Set up a universe using the roulette wheel selection mechanism for selecting a white hole based on the universe inflation rate.

Step 3 Update the Travel Distance rate (TDR), and Wormhole Existence Probability (WEP) and check the boundaries. The probability of wormhole existence in the universe is determined using two above-mentioned coefficients. By increasing the linearity over iterations, exploitation is getting more emphasis as the optimization progresses. Also, the distance rate (variation) can be also defined by TDR. This rate determines the distance that an object can teleport through a wormhole around the best universe at the moment. In fact, more accurate exploitation (local search) is realized around the best-obtained universe by having TDR to increase over iterations.

$$WEP = \min + l \cdot \left(\frac{max - min}{L} \right) \tag{6}$$

$$TDR = 1 - \frac{l^{1/p}}{l^{1/p}} \tag{7}$$

where Min and Max show the minimum and maximum WEPs respectively. Furthermore, I, L, and p denotes the current iteration, the maximum number of iterations, and the exploitation accuracy respectively. Totally, it can be declared that a low WEP with a high TDR support exploration whereas a high WEP with a low TDR support exploitation. To get desirable search results, it is very important to make a compromise between two opposing forces. Obviously, the fitness values, WEPs, and TDRs change in each iteration. In a simple word, in the universe with the best fitness value, TDR should increase and WEP should decrease. However, in other universes, TDR should decrease and WEP should increase. It is a vivid fact that WEP and TDR are the most important and influential parameters that play the role of exploration and exploitation during optimization. For this reason, after each iteration, the existence probability of wormholes smoothly increases (WEP values increase) in order to emphasize more on exploitation during the optimization process. However, at the same iteration time, the traveling distance of variables decreases (TDR values decrease).

Step 4 Calculate the current inflation rate of the universe. When the inflation rate of the universe outperforms its current inflation rate, the current one should be updated. Otherwise, one should maintain the current one.

Step 5 Update the universe position according to the following equation.

$$x_i^j = \begin{cases} \begin{cases} X_j + TDR \times \left(\left(ub_j - lb_j\right) \times r4 + lb_j\right) & r3 < 0.5 \\ X_j - TDR \times \left(\left(ub_j - lb_j\right) \times r4 + lb_j\right) & r3 \geq 0.5 \end{cases} & r2 < WEP \\ x_i^j & r2 \geq WEP \end{cases} \tag{8}$$

where X_j denotes the *j*th parameter of the best universe found so far. Also, ub_j and lb_j define the upper and lower bounds of *j*th variable respectively. Also, x_i^j denotes the *j*th parameter of *i*th universe. Finally, *r*2, *r*3, *r*4 denotes the random numbers taken from the interval of [0, 1].

Step 6 Terminate the algorithm. By realization of the termination criterion, the algorithm stops and introduces the corresponding result as final output. Otherwise, the number of iterations increases by 1 and the algorithm returns to Step 2.

Comparing with other metaheuristic algorithm algorithms, MVO provides a strong ability in the optimization process with regard to fewer control parameters. In fact, the optimization process begins with initializing a set of universes with random numbers. During each iteration, variables in the universes with a high inflation rate move toward the universes with low inflation values by way of white or black holes. Every universe runs into random theoretical transfer in its variables through wormholes towards the best universe. This process is iterated until a pre-defined maximum number of iterations. Furthermore, the MVO algorithm preserves the best solution during optimization.

4 Proposed MVO-GBDT algorithm

In the following, three significant points about the proposed MVO are described to get a better feature selection and GBDT parameter optimization. Indeed, we explain the fitness function, system architectures, and encoding scheme employed for the representation of MVO universes in the following.

4.1 Encrypted plan (structure of solution)

In this work, a vector of real numbers encodes the individuals so that the number of features in the dataset plus four is equal to the number of elements in a vector. In fact, four elements are used to represent the parameters of GBDT namely, learning rate, a tree's maximum depth, number of estimators, and minimum number of leaf nodes. Figure 2 shows the encoding scheme implemented in this work. The numbers randomly generated in the interval [0, 1] are used as the elements of the vector. Then, the elements larger than or equal to 0.5 are rounded to 1 (so, the feature is selected); otherwise, they are rounded to zero and the feature is discarded. Because of different search spaces, we should map the parameters of GBDT into different scales.

For instance, we map the element corresponding to the number of estimators into the interval [1, 200] while the interval [1, 32] is taken into account for the element representing a tree's maximum depth. The following equation is employed to transform the values of the parameters in a linear manner.

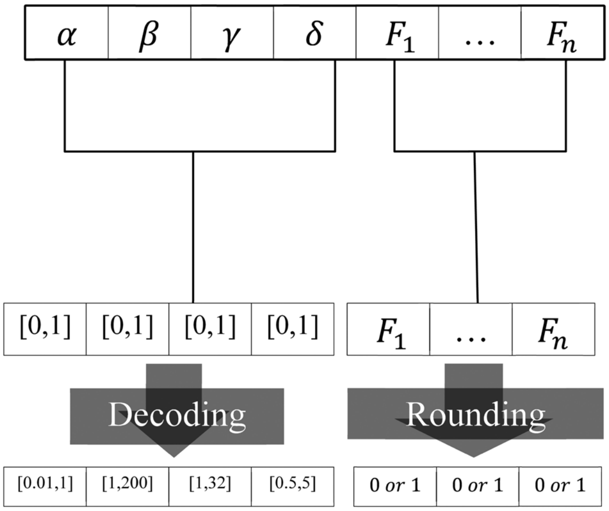


Fig. 2 Encoding scheme of solutions for GBDT parameter tuning and feature selection (α: learning rates, β: number of estimators, γ: max depths, δ: min samples leafs)

$$\text{output} = \frac{input - min_{input}}{max_{input} - min_{input}}\left(max_{output} - min_{output}\right) + min_{output}. \quad (9)$$

4.2 Objective function

With accuracy calculation for each selection, each solution can be evaluated by objective function. It should be noted that the confusion matrix (widely used for classification evaluation) is employed to calculate the accuracy. In fact, after splitting data into ten different folds, nine out of ten folds are trained and the first fold is tested by model and the accuracy of model is calculated by a confusion matrix representing the classification of 1/10 of the data. This process continues for the next test set (the second fold) to get another confusion matrix for another 1/10 of the data. Finally, this process stops when all folds are tested. Accordingly, the performance of the model is obtained by summing all the calculated confusion matrices. In fact, it is the skill of a model on new data that is estimated by k-fold cross-validation method. The advantages of this method are bias reduction and using all the data for model evaluation. A confusion matrix is also known as an error matrix. In the field of ML (particularly, statistical classification problems), an error matrix is a specific table layout making algorithm performance visualization possible. A confusion matrix has shown in the following Table 1.

According to the confusion matrix, the accuracy rate of the classification can be calculated as follows.

$$Accuracy = \frac{T_P + T_N}{T_P + F_N + F_P + T_N} \times 100\% \quad (10)$$

where T_P denotes the number of correct predictions and actual class is true. Also, T_N denotes the number of correct predictions and actual class is false. Furthermore, F_N and F_p denote the number of incorrect predictions with true and false actual classes respectively. Moreover, each specific solution corresponds a specific model with specific tuned parameters and features. Due to the fact that a solution with high classification accuracy and a small number of selected features is better than other solutions, all of these mentioned factors must be taken into account to design the fitness function. The following multi-objective function is employed to calculate the fitness value.

$$Fitness = (\alpha \times mean(Accuracy)) - \left(\beta \times \frac{|f_s|}{|f_T|}\right) - (\gamma \times std(Accuracy)) \quad (11)$$

where $mean(Accuracy)$ is mean of classification accuracy of the ten outputs obtained from the tenfold cross-validation, $|f_s|$ is the number of selected features, $|f_T|$ is the number of total features and $std(Accuracy)$ is the standard deviation of classification accuracy for all the ten outputs obtained from the tenfold cross-validation. In general, the solution with higher fitness value introduces a more efficient model. Given that the smaller the number of features, the less data is computed and also the lower standard deviation of the accuracies of the ten outputs obtained from the tenfold cross-validation indicates more stable performance for the model, these two factors are considered in Eq. (11) as a penalty for the accuracy of the classification model.

Table 1 Confusion matrix

Confusion matrix	Predicted class	
	Positive	Negative
Actual class		
Positive	T_P	F_N
Negative	F_P	T_N

4.3 System architecture

This section is used to describe the proposed MVO-GBDT system architecture. Moreover, in this section, we use the MVO population-based algorithm to optimize the parameters of the GBDT. In the algorithm, a vector of real numbers is obtained by encoding individuals (universes). The number of the elements in each vector is equal to the number of the features in the dataset plus four elements representing GBDT parameters. The main parts of the proposed system architecture are shown in Fig. 3 and also described in the following.

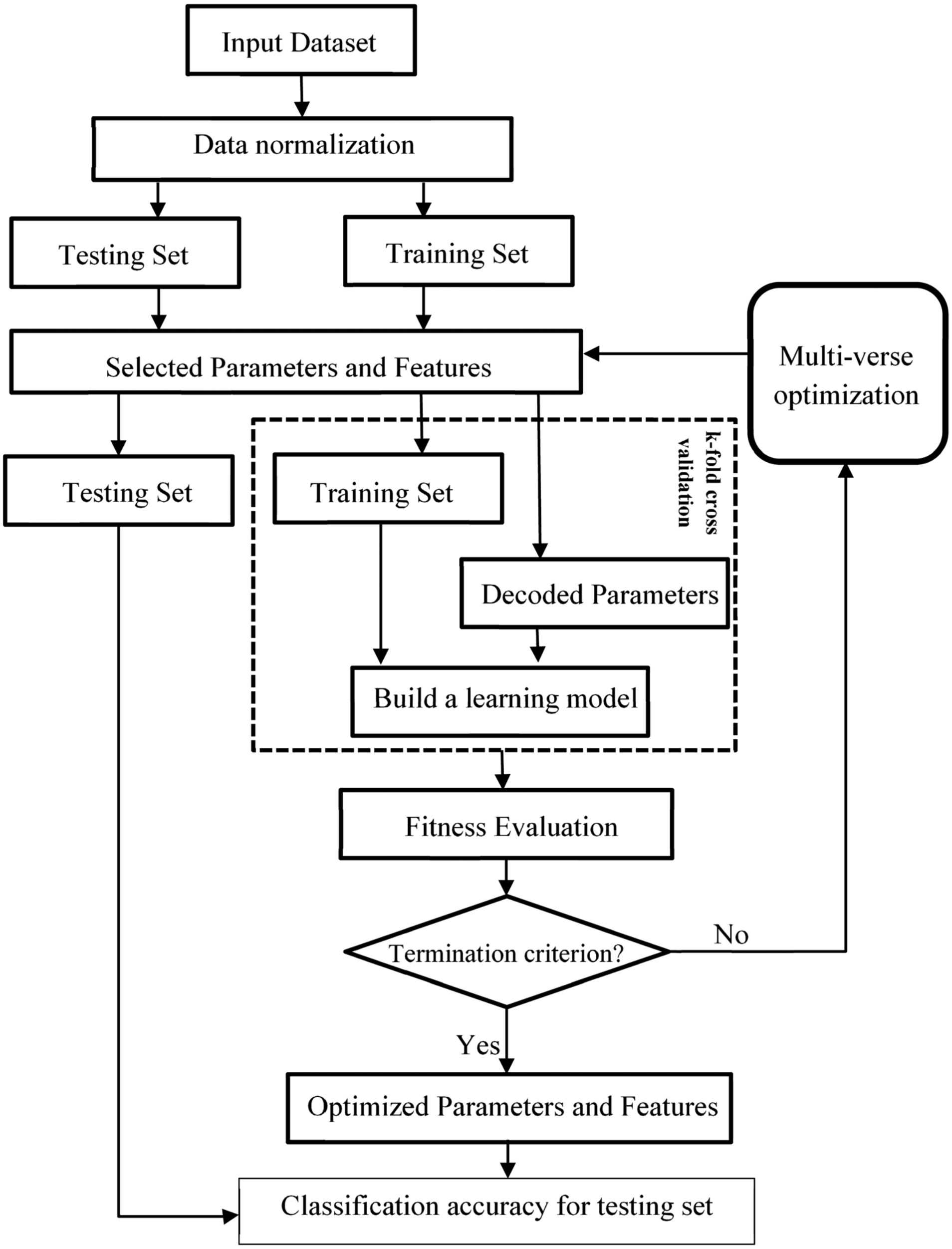
Step 1 Data normalization [45]. This step eliminates the effect of the different range values of features on the learning process. For this purpose, we map the values of all features on the same scale. Various normalization methods have been developed in researches for data re-scaling. Thus, equal weight is determined for all features to normalize them in the interval [0, 1] as follows.

$$X_{new} = \frac{X - Min_A}{Max_A - Min_A} \quad (12)$$

where Min_A and Max_A show the minimum and maximum values of a given attribute respectively. Also, X_{new} shows the mapped value of X that takes a value between 0 and 1.

Step 2 divides the normalized data into training and testing sets. Then, the training part is split again into a number of smaller parts using k-fold cross-validation.

Fig. 3 System architecture of the GBDT–MVO approach



So, by training SVM in k steps, the average evaluation is used again. The training data is used for model estimation and validation data is employed for final model selection. Finally, the final model is employed to test and compare other models.

Step 3 initializes the considered parameters namely Max Iteration (the maximum number of iterations), the universe number, and the range of candidate values for each object of an individual solution. In fact, a set of GBDT parameters and features of data represent an individual solution.

Step 4 initializes the universe position. Each universe in the MVO algorithm represents a set of (learning rate, number of estimators, maximum depth of a tree, min leaf nodes,$f_1,\ldots,f_n$). This set is initialized based on the parameter range in the previous step.

Step 5 decodes the universes. The vectors (universes) obtained by MVO are divided into two parts. To describe precisely, the first four elements (corresponding to the parameters of GBDT parameters) are converted using Eq. (9) and the rest (corresponding to the selected features) are rounded to make a binary vector.

Step 6 selects the feature subset. After obtaining the binary vector in the previous step, the corresponding features are selected from the training dataset.

Step 7 evaluates the fitness values. Every solution generated by MVO can be evaluated by taking advantage of the described fitness function. Using a confusion matrix for a binary classifier, the accuracy of trained model is evaluated based on the generated parameters. After calculating the accuracy of a trained model and sorting the universes, a white hole is selected by roulette mechanism. It should be noted that the evaluation criterion employed in the current study aims to identify the GBDT model's suitable parameters. In fact, the higher the accuracy of a model, the better is its performance.

Step 8 updates the WEP and TDR based on the Eqs. (6) and (7).

Step 9 update current fitness. If the fitness of the universe is better than the current fitness, the algorithm will update the current fitness of the universe. Otherwise, no action is needed.

Step 10 updates the universes' positions and finds the optimal individual in the optimal universe.

Step 11 terminates the algorithm. By realization of the termination criterion, the algorithm stops and introduces the corresponding result as final output. Otherwise, the number of iterations increases by 1 and the algorithm returns to Step 2.

The computational complexity of the GBDT–MVO reliant on the computational complexity of MVO and GBDT. The GBDT has computational complexity of O $(n \times p \times n_{trees})$. While n is the number of training sample, p the number of features, n_{trees} is the number of trees. The computational complexity of the MVO algorithms relay on the maximum number of iterations (l), number of universes (m), the number of objects (d), roulette wheel mechanism, and universe sorting algorithm (quicksort algorithm). Since the roulette wheel selection is used for every object in every universe over the iterations, roulette wheel mechanism has the complexity of $O(m)$. In addition, quicksort algorithm has the complexity of $O(\mathrm{m} \times \log m)$ and $O(m^2)$ in the best and worst case, respectively. Therefore, the overall computational complexity is as follows:

$$O(GBDT - MVO) = l \times \left(O\left(m^2\right) + m \times d \times O(m) + m \times O(n \times p \times n_{trees})\right) \quad (13)$$

Figure 3 shows the workflow of the GBDT–MVO approach and the relationships among the main system parts.

5 Experimental results and analysis

5.1 Data description

In this work, two standard datasets (from the University of Wisconsin Hospitals, Madison) [46–49] are employed to evaluate the proposed GBDT–MVO approach for breast cancer diagnosis. In the following, we describe these datasets briefly.

5.1.1 Wisconsin Original Breast Cancer (WBC) Dataset

In this dataset, all features are obtained from a digitized image of a breast mass's fine needle aspirate (FNA). In fact, these features describe the characteristics of the cell nuclei existing in the image. The target feature records the prognosis (cancerous or non-cancerous). Also, all samples are periodically updated by the reports of Dr. Walberg's clinical cases. In addition, this data set consists of 10 features and 699 instances, including a patient ID and other features indicated in Fig. 4. This figure shows the correlation matrix of the dataset features, determining the correlation coefficients between variables.

In Fig. 4, each cell in the table represents the correlation between the two variables. The correlation coefficient is a statistical relationship between two variables. The values range between +1 (perfect direct relationship) and −1 (perfect inverse relationship). Also, a correlation of 0 shows no relationship between the movement of the two variables. Moreover, Table 2 lists a summary of each attribute's range.

5.1.2 Wisconsin diagnostic breast cancer (WDBC) dataset

In this dataset, all features are computed based on a digitized image of a breast mass's FNA. Relevant features were selected by searching the feature space thoroughly and separating planes. It should be noted that this dataset consists of 32 features and 569 instances (62.74% cancerous and 37.26% non-cancerous), including a patient ID, 30 tumor features, and one class indicator in the WDBC dataset. The correlation matrix of this dataset is shown in the following Fig. 5.

The aspects considered in tumor feature collection are texture, radius, perimeter, area, compactness, smoothness, concavity, symmetry, concave points, and fractal dimension. Table 3 lists a summary of each attribute's range.

5.2 Performance evaluation methods

The performance evaluation approaches employed to assess the proposed GBDT–MVO method are presented

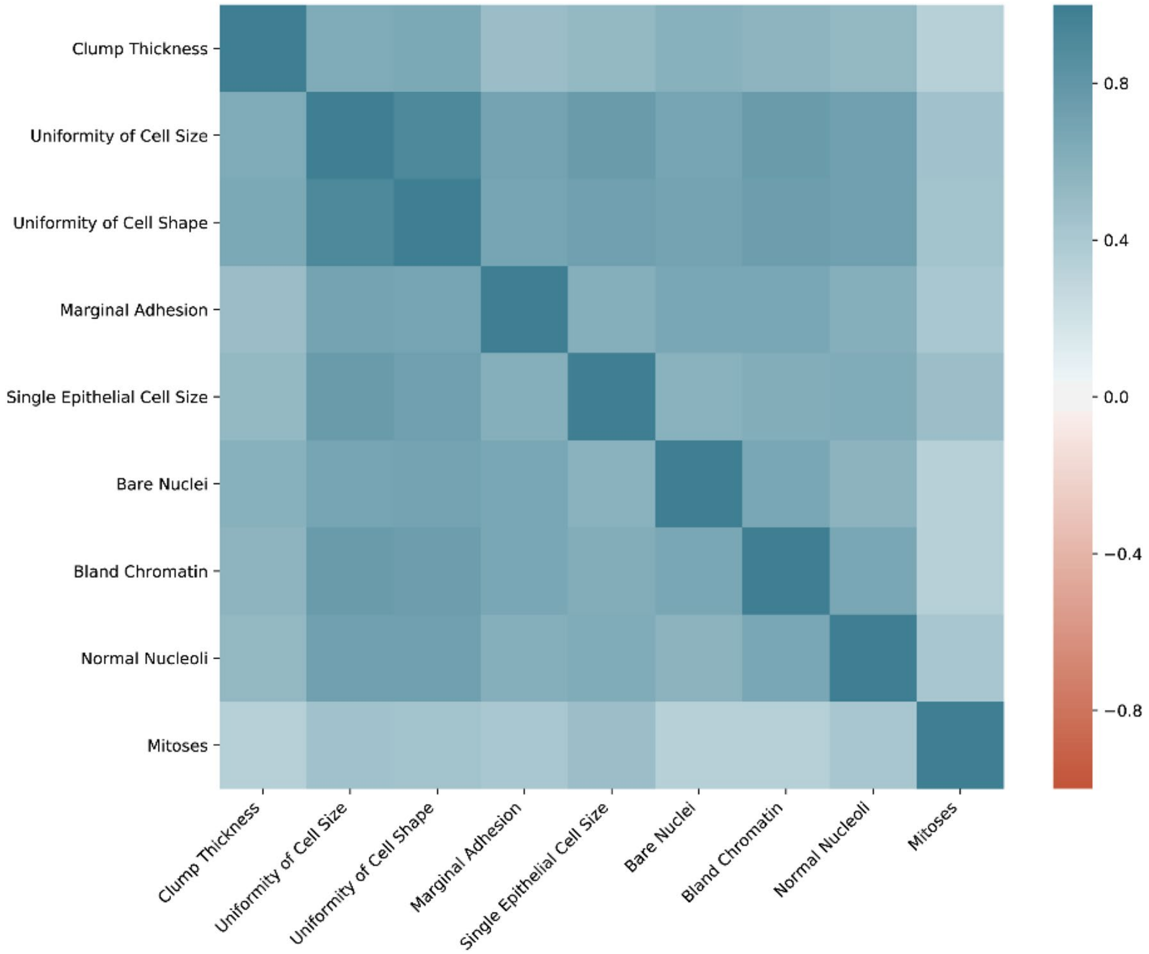


Fig. 4 Feature correlation of WBC dataset

in the following. In this regard, the confusion matrix was employed to evaluate the performance of the classifiers for breast cancer detection. The other criteria employed for evaluation are specificity, sensitivity, Matthews correlation coefficient (MCC), F-measure, and the area under the Receiver Operating Characteristic curve (AUC) [50–52].

Table 2 summary of WBC dataset attribute's range

Attributes	Domain	Mean	SD
Clump thickness	1–10	4.44	2.82
Uniformity of cell size	1–10	3.15	3.07
Uniformity of cell shape	1–10	3.22	2.99
Marginal adhesion	1–10	2.83	2.86
Single epithelial cell size	1–10	3.23	2.22
Bare nuclei	1–10	3.54	3.64
Bland chromatin	1–10	3.45	2.45
Normal nucleoli	1–10	2.87	3.05
Mitoses	1–10	1.60	1.73

Furthermore, the performance indices used for evaluation and comparison are as follows.

$$Sensitivity = \frac{T_P}{T_P + F_N} \times 100\% \tag{14}$$

$$Specificity = \frac{T_N}{F_P + T_N} \times 100\% \tag{15}$$

$$Precision(P) = \frac{T_P}{T_P + F_P} \tag{16}$$

$$Recall(R) = \frac{T_P}{T_P + F_N} \tag{17}$$

$$F{-}measure = \frac{2 \times P \times R}{P + R} \tag{18}$$

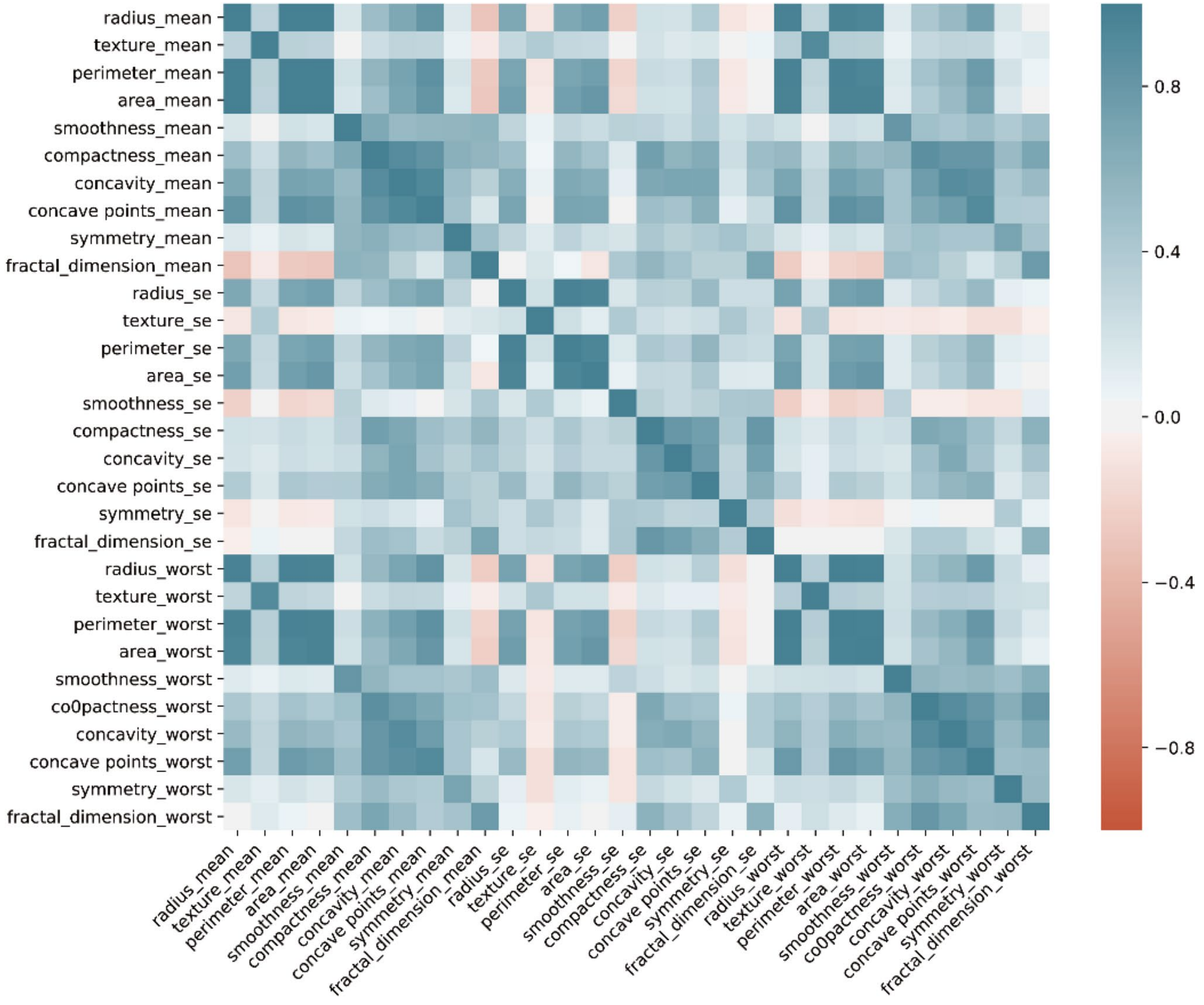


Fig. 5 Feature correlation of WDBC dataset

Table 3 summary of WDBC dataset attribute's range

Attributes	Range		
	Mean	Standard error	Largest value
Radius	6.98–28.11	0.11–2.87	7.93–36.04
Texture	9.71–39.28	0.36–4.89	12.02–49.54
Perimeter	43.79–188.50	0.76–21.98	50.41–251.20
Area	143.50–2501.00	6.80–542.20	185.20–4254.00
Smoothness	0.05–0.16	0.00–0.03	0.07–0.22
Compactness	0.02–0.35	0.00–0.14	0.03–1.06
Concavity	0.00–0.43	0.00–0.40	0.00–1.25
Concave points	0.00–0.20	0.00–0.05	0.00–0.29
Symmetry	0.11–0.30	0.01–0.08	0.16–0.66
Fractal dimension	0.05–0.10	0.00–0.03	0.06–0.21

$$MCC = \frac{T_P \times T_N - F_P \times F_N}{\sqrt{(T_P + F_P)(T_P + F_N)(T_N + F_P)(T_N + F_N)}} \tag{19}$$

According to above Equations, basic performance measures were derived from the confusion matrix. Accordingly, four outcomes were obtained by a binary classifier. In this regard, the F_P, T_P,F_N, and T_N measures can collectively build a plot called Receiver Operating Characteristic (ROC) curve. This curve represents the trade-off between FN and FP rates and model classification errors. As seen in this figure, in ROC curves, FP rate is typically plotted versus TP rate. Also, AUC can be obtained according to the ROC curve. In other words, ROC, as a probability curve, results in AUC representing the measure or degree of separability. In fact, the higher the AUC the higher the model's classifying ability. An AUC near to one shows a supreme model with great separability measure. On the contrary, an AUC near to zero implies a very bad separability measure. Supposing that sensitivity and (1-specificity) are the probabilities of T_P, and F_P respectively, AUC can be estimated as follows.

$$AUC = \sum_i \left\{ \left[Sensitivity_i \cdot \Delta(1{-}Specificity)\right] + \frac{1}{2}\left[\Delta Sensitivity \cdot \Delta(1{-}Specificity)\right] \right\} \tag{20}$$

Where $\Delta(1{-}Specificity) = (1 - Specificity)_i - (1 - Specificity)_{i-1}$ and $\Delta Sensitivity = Sensitivity_i - Sensitivity_{i-1}$. Here i is used as an index. The experimental results are presented and discussed in the next section. It should be noted that the described performance evaluation methods described methods were used to assess the ability of the proposed method in breast cancer diagnosis.

5.3 Experiments setup

In this study, the above-mentioned breast cancer datasets were used to evaluate algorithms. Each instance in these datasets includes the class attribute which has four values (such as Malignant and Benign). Also, the classifiers were analyzed using tenfold cross-validation and compared by taking advantage of Scikit-Learn libraries in the Python programming language. The device employed to run experiments was a PC with an Intel Core i5-2.20 GHz CPU and 16 GB RAM.

5.4 Results and discussion

To evaluate the performance of our proposed method, we considered the effectiveness and reliability improvements compared to previous methods.

During all experiments, the generalization errors of obtained models were estimated using K-fold cross-validation [53]. In fact, all models were trained based on K − 1 partitions and tested using the *K*th partition to obtain the testing performance P_k. According to the following equation, the overall performance is the average of the performances resulted from K iterations.

$$\bar{P} = \frac{1}{K}\sum_{k=1}^{K} P_k \tag{21}$$

where P_k is a performance measure to evaluate the diagnostic accuracy in a different way and was used to test the performance of the proposed model (i.e. the accuracy, sensitivity, specificity, F-measure, and AUC) [54].

Table 4 lists the initial parameters employed in MVO, PSO, GA, and BAT algorithms. As seen, the swarm size and the number of universes and individuals are similar (50) in all the mentioned algorithms. In addition, the same number of iterations (15) is set for each of algorithms in order to fairly compare all the metaheuristic methods.

Table 4 Initial parameters of the MVO, GA, PSO and BAT

Algorithm	Parameter	Value
MVO	Min wormhole existence rate	0.2
	Max wormhole existence rate	1
	Iterations	15
	Universes	50
BAT	Loudness	0.5
	Pulse rate	0.5
	Number of artificial bats	50
	Iterations	15
	Frequency minimum	0
	Frequency maximum	1
GA	Crossover rate	0.8
	Mutation rate	0.1
	elitist ratio	0.1
	Selection mechanism	Roulette wheel
	Population size	50
	Generations	15
PSO	Acceleration constants	[1.5, 1.7]
	Inertia weight	1
	Generations	50
	Number of particles	15

Table 4 presents the initialize setting of various parameters in the compared and the proposed approach which leads to a direct effect on the algorithm performance. It is very important to take into account some considerations when determining the maximum number of algorithm iterations. In fact, the small number of iterations helps to prevent the over-fitting issue and high computing costs. Also, it mitigates the time required by metaheuristic algorithms for computing and boosts their convergence.

In this part of the experiments, MVO was evaluated and compared to GA, PSO, and BAT algorithms in terms of feature selection and GBDT's parameters optimization. Moreover, all four population-based approaches worked based on our proposed system architecture and also these approaches were implemented and evaluated using the two datasets described earlier. Tables 5 and 6 presents the average accuracy rate and the average number of selected features respectively.

In Table 5, the results of BCW dataset indicate that MVO has achieved the highest average accuracy rates compared to other algorithms. This Table shows that BAT and PSO have achieved a 96.7% accuracy rate which is not really close to MVO with an accuracy of 97.13%. Also, one can realize that the numbers of the selected features in three optimizers do not differ significantly.

The results on WDBC dataset presented in Table 6 show that BAT and PSO have achieved approximately same accuracy rate (98.59% and 98.06% respectively). Moreover, this Table shows that MVO has achieved the highest average accuracy rates compared to GA, PSO, and BAT. As seen, the three optimizers have had significant results in terms of the number of selected features. In another experiment, our proposed MVO was compared with the gird search in terms of GBDT's parameters optimization. Considering the fact that the grid search is not able to perform feature selection, MVO was applied just for parameter optimization to compare them fairly. Tenfolds cross-validation was used for both approaches. The comparison results are presented in Table 7. According to the results, MVO obviously outperforms the grid search in both datasets. As can be seen, in the case of the BCW dataset, MVO is slightly better. However, in the case of the WDBC dataset, MVO considerably outperforms the grid search in terms of obtained accuracy rates.

It is undeniable that model reliability is very important and effective in disease diagnosis. In fact, reliable model provides high diagnostic accuracy with durable stability. To determine the reliability improvement of the ensemble technique, we used the performance variance. It is worth noting that since we want to compare the reliability of the ensemble models, the variance should be measured based on standard deviation (σ) as follows.

$$\sigma(P) = \frac{1}{L-1}\sum_{l=1}^{L}(\bar{P}_l - \bar{P}) \tag{22}$$

where L and denotes the number of replications and P is a performance measure like accuracy or AUC [54].

Figures 6, 7 and 8 indicate the variance of accuracy and AUC using box plots and ROC curves (for both datasets) respectively.

The ROCs of all optimizers are shown in Fig. 6 based on the AUC rates. As seen in this figure, MVO has the highest rate. The average and standard deviations of AUC are shown by *avg* and *std* respectively.

In the following experiment, we used the variance of accuracies (shown by box plots in Fig. 8) to study the reliability improvement of the ensemble technique. Figure 8 indicates variability outside the upper and lower quartiles concerning K-fold cross-validation results. In other words, Box plots illustrate the variation in the accuracy of tenfold cross-validation results. In Fig. 8a, GBDT–MVO accuracy ranged from 0.95 to 1, GBDT-PSO ranged from 0.91 to 1, and GBDT (Grid Search) ranged from 0.91 to 0.98. Moreover, Fig. 8b, GBDT–MVO reach

Table 5 Results of presented system architecture (Wisconsin original breast cancer dataset)

Model	No. of selected features	Accuracy	Specificity	Sensitivity	F-measure	MCC
MVO + GBDT	4	0.9713	0.9716	0.9709	0.9590	0.9372
GA + GBDT	6	0.9513	0.9585	0.9377	0.9300	0.8928
PSO + GBDT	5	0.9670	0.9694	0.9626	0.9527	0.9276
BAT + GBDT	5	0.9670	0.9716	0.9585	0.9525	0.9274

Table 6 Results of presented system architecture (Wisconsin diagnostic breast cancer dataset)

Model	No. of selected features	Accuracy	Specificity	Sensitivity	F-measure	MCC
MVO + GBDT	6	0.9876	0.9764	0.9943	0.9902	0.9736
GA + GBDT	9	0.9841	0.9716	0.9915	0.9874	0.9661
PSO + GBDT	6	0.9806	0.9622	0.9915	0.9847	0.9586
BAT + GBDT	7	0.9859	0.9716	0.9943	0.9888	0.9699

Table 7 Best obtained results without feature selection

Dataset	Model	Accuracy	Specificity	Sensitivity	F-measure	MCC
Breast cancer wisconsin (BCW)	MVO	0.9699	0.9737	0.9626	0.9567	0.9337
	Grid search	0.9628	0.9694	0.9502	0.9462	0.9178
Wisconsin diagnostic breast cancer (WDBC)	MVO	0.9771	0.9575	0.9887	0.9819	0.9510
	Grid search	0.9367	0.9245	0.9439	0.9492	0.8652

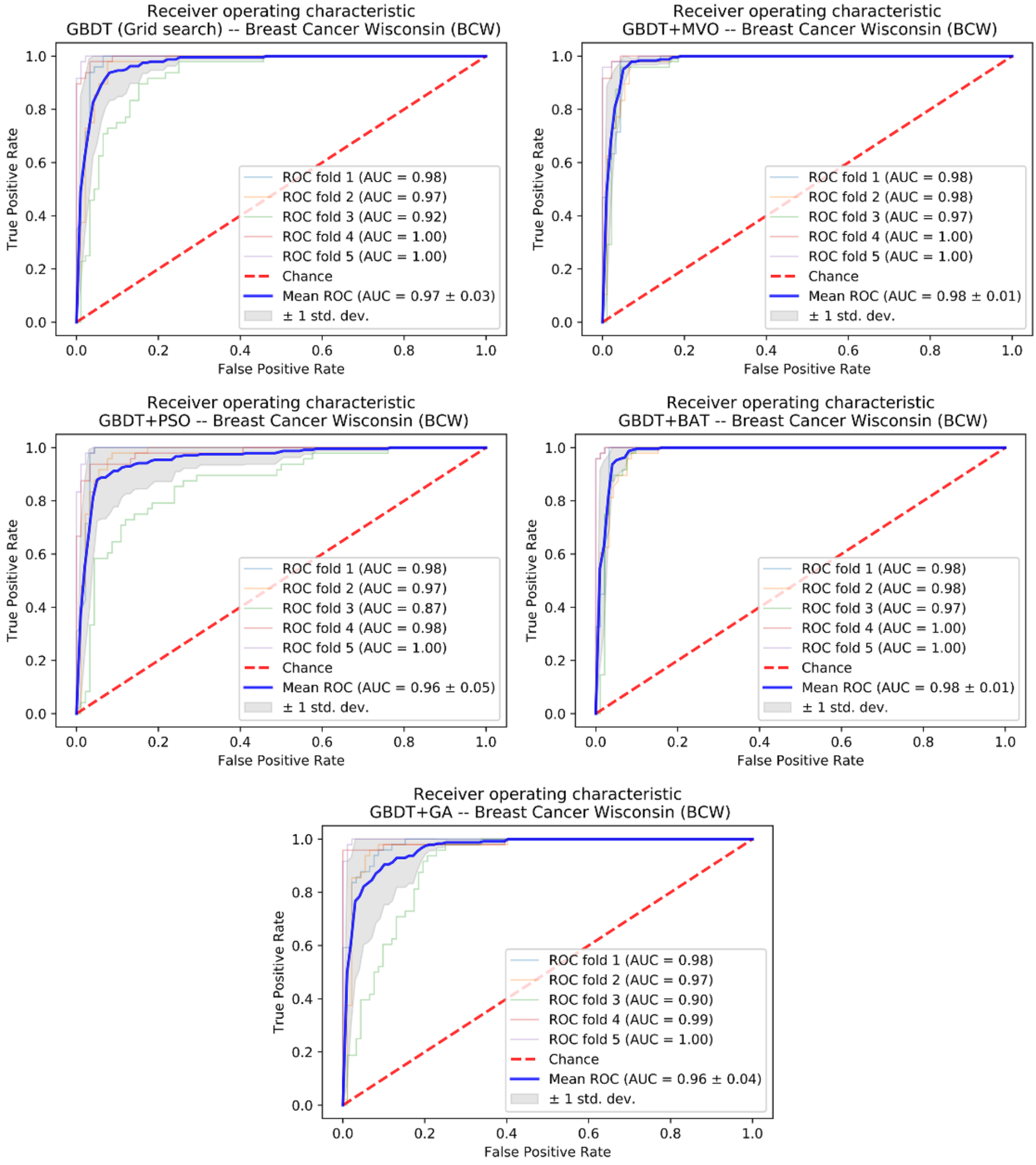


Fig. 6 Comparison of ROC with different models (BCW dataset)

a narrow accuracy range with a higher median regarding other methods which indicate performance robustness of the presented method. The comparison results demonstrated that our proposed approach is able to provide high accuracies with small diagnosis variance. Also, these results proved that our initial objective was realized—improved diagnosis accuracy with reduced diagnosis variance. As illustrated in Fig. 8, GBDT–MVO boxplot follows a normal distribution (symmetric) which shows that the distribution of accuracies obtained from

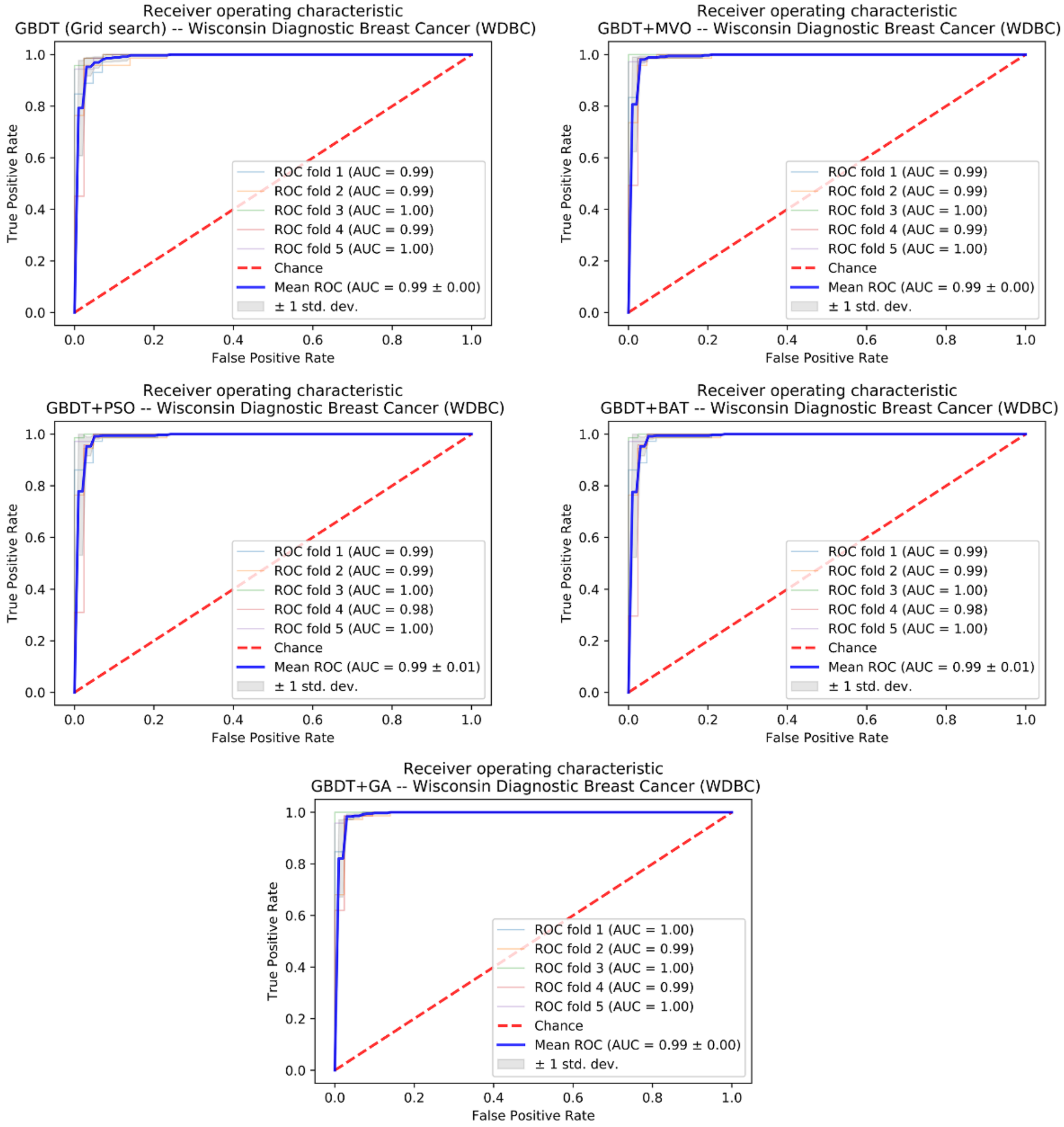


Fig. 7 Comparison of ROC with different models (WDBC dataset)

K-fold cross-validation is not skewed. In fact, the whiskers of other boxplot visualize outliers in a clear way. In fact, this Figure shows the performance robustness of the proposed method.

According to the above-mentioned discussions, it can be concluded that when new data are added to the model, MVO considerably outperforms the grid search and other population-based approaches in terms of GBDT optimization.

Moreover, Fig. 9 illustrates a comparison with other studies available in the literatures (ACO-SVM [55], GA-SVM [55], PSO-SVM [55], Naive Bayes(NB) [56], Multi-Layer Perception [56], Decision tree (J48) [56], Hybrid of K-means and SVM [57], BP neural network [58], WAUCE model [4]) based on WDBC dataset. In this comparison,

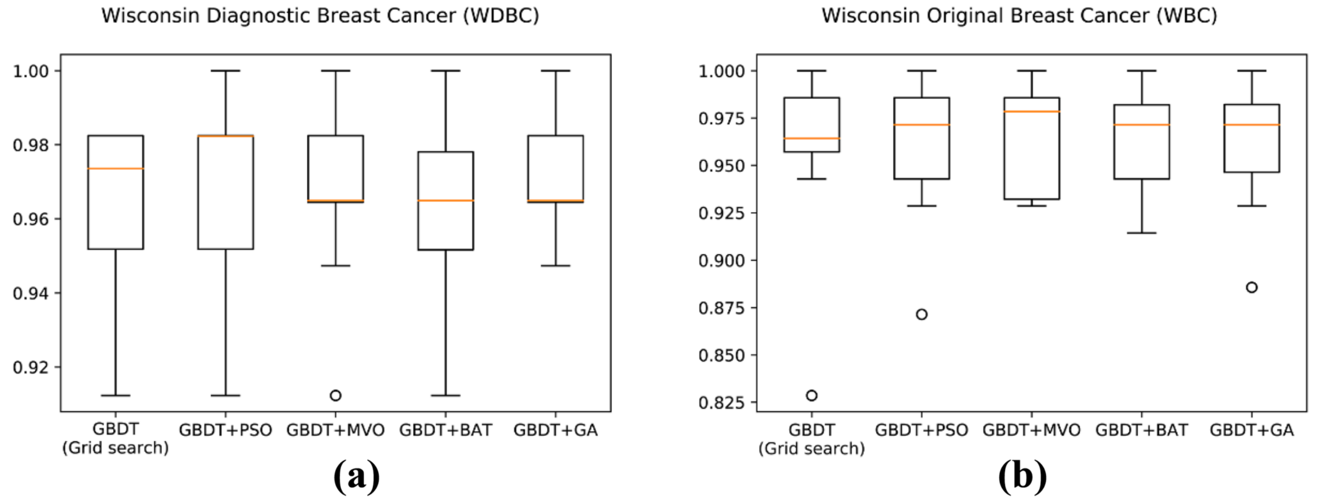
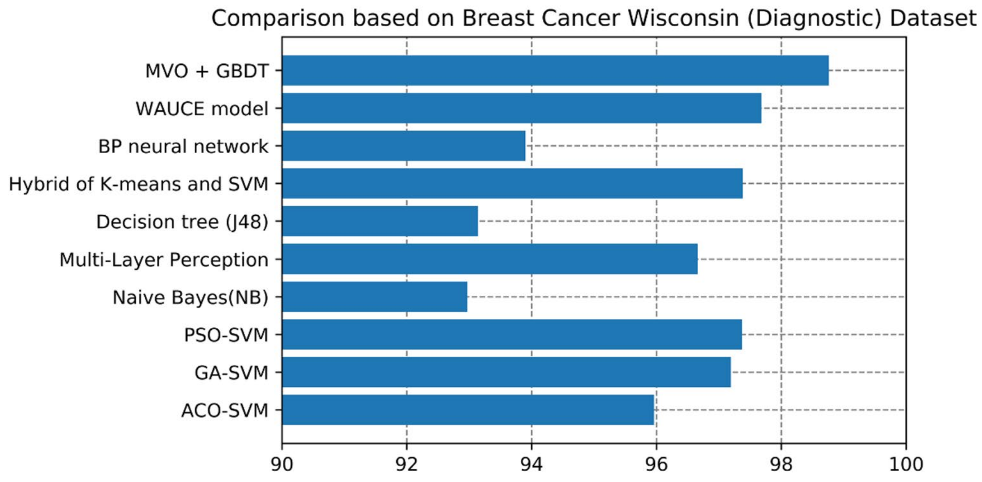


Fig. 8 Results of experiment without feature selection (diagnosis variance illustration)

Fig. 9 Comparison of GBDT + MVO with other studies in terms of accuracy



tenfold cross-validation was adopted as a standard evaluation criterion.

One of the significant challenges that learning models have to deal with is high dimensional and features of data. In addition, the majority of data from real-world applications associated with extremely redundant data. For this reason, GBDT–MVO aims to reduce the problem of high dimensionality regarding redundant data and eliminating the features with a low correlation. According to the results presented in the previous section, since our proposed MVO algorithm has high exploitation ability, it was successful and improved the performance of GBDT by tuning the related parameters precisely. Also, since wormholes had a significant role in obtaining best individual, the MVO algorithm was very effective in quality improvement of solutions. In addition, the high exploration ability of MVO avoided local optima traps and improved the MVO-based GBDT's reliability and robustness. It should be noted that the role of white holes and black holes was very bold in making sudden changes in the solutions and avoiding local optima traps. Furthermore, by taking advantage of TDR and WEP parameters, MVO was able to first explore the search space thoroughly and then exploit the high-potential regions precisely over the iterations. This perfect trade-off between exploration and exploitation led the MVO algorithm to outperform other algorithms.

To sum up, the results and findings of this section indicate that using the presented fitness function with regard to the MVO algorithm is able to improve the performance of GBDT. Since the GBDT performance reliant on the parameters that must be tuned accurately, MVO is able to do the tuning process very efficiently. GBDT–MVO reached the highest accuracy overall considered data sets. In fact, the combination of GBDT with MVO provides more accurate results. The results indicate the stability which came from the MVO robustness in the convergence rate in complicated search spaces. However, one of the most important factors affecting the use of the proposed method in other datasets is determining the appropriate values for the MVO parameters. Because the speed and accuracy of optimization process (the feature selection and parameter tuning of GBDT) depends on the number of universes and the number of iterations. Besides, most of the metaheuristic has close time complexity (linear complexity), the run time complexity is approximately the same as other compared methods. The runtime analysis of current data sets indicates that the runtime for each considered method is in the range of 0.46–0.92 s which is insignificant because of the small number of instances.

6 Conclusions

In the field of breast cancer diagnosis, the ML-based and soft-computing-based medical decision support systems have been very efficient and effective. In this study, by taking advantage of ensemble learning, we combined the GBDT and MVO to propose a robust classifier for optimal classification of datasets. Here, WDBC and BCW datasets were employed as standard breast cancer datasets to show the high reliability and effectiveness of the proposed model. As mentioned earlier, this study's main objective was to improve the breast cancer classification accuracy. The obtained results demonstrated that cancer diagnosis performance can be significantly enhanced using the proposed model. Also, we compared the performance variances to show that how important is the model reliability to diagnose diseases. In fact, the proposed model was able to decrease the diagnosis accuracy variance while increasing accuracy. In total, the experimental results and subsequent discussions proved that the proposed ensemble method outperforms other existing methods in this field and the breast cancer classification performance can be noticeably improved using the proposed classifier.

Compliance with ethical standards

Conflict of interest The authors declare that they have no conflict of interest.

Ethical approval This article does not contain any studies with human participants or animals performed by any of the authors.

Informed consent Informed consent was obtained from all individual participants included in the study.

Human and animal rights This article does not contain any studies with human participants or animals performed by any of the authors.

References

1. Singh BK (2019) Determining relevant biomarkers for prediction of breast cancer using anthropometric and clinical features: a comparative investigation in machine learning paradigm. Biocybern Biomed Eng 39(2):393–409. https://doi.org/10.1016/j.bbe.2019.03.001
2. Patri A, Patnaik Y (2015) Random forest and stochastic gradient tree boosting based approach for the prediction of airfoil self-noise. Procedia Comput Sci 46:109–121. https://doi.org/10.1016/j.procs.2015.02.001
3. Kaur P, Singh M, Josan GS (2015) Classification and prediction based data mining algorithms to predict slow learners in education sector. Procedia Comput Sci 57:500–508. https://doi.org/10.1016/j.procs.2015.07.372
4. Wang H, Zheng B, Yoon SW, Ko HS (2018) A support vector machine-based ensemble algorithm for breast cancer diagnosis. Eur J Oper Res 267(2):687–699. https://doi.org/10.1016/j.ejor.2017.12.001
5. Korkmaz SA, Poyraz M (2015) Least square support vector machine and minumum redundacy maximum relavance for diagnosis of breast cancer from breast microscopic images. Procedia Soc Behav Sci 174:4026–4031. https://doi.org/10.1016/j.sbspro.2015.01.1150
6. Naga RamaDevi G, Usha Rani K, Lavanya D (2018) Ensemble-based hybrid approach for breast cancer data. ICCCE 2018:713–720. https://doi.org/10.1007/978-981-13-0212-1_72
7. Aličković E, Subasi A (2015) Breast cancer diagnosis using GA feature selection and rotation forest. Neural Comput Appl 28(4):753–763. https://doi.org/10.1007/s00521-015-2103-9
8. Mandal I (2012) Enhanced breast cancer recognition based on rotation forest feature selection algorithm. Comput Sci Inf Technol (CS & IT). https://doi.org/10.5121/csit.2012.2322
9. Nguyen C, Wang Y, Nguyen HN (2013) Random forest classifier combined with feature selection for breast cancer diagnosis and prognostic. J Biomed Sci Eng 06(05):551–560. https://doi.org/10.4236/jbise.2013.65070
10. Bhardwaj A, Tiwari A (2015) Breast cancer diagnosis using genetically optimized neural network model. Expert Syst Appl 42(10):4611–4620. https://doi.org/10.1016/j.eswa.2015.01.065
11. Bhardwaj A, Tiwari A, Chandarana D, Babel D (2014) A genetically optimized neural network for classification of breast cancer disease. In: 2014 7th International conference on biomedical engineering and informatics. https://doi.org/10.1109/bmei.2014.7002862
12. Kaya Y (2015) A hybrid model for breast cancer diagnosis based on expection-maximization and artificial neural network: EM + ANN. Karaelmas Sci Eng J 5(1):26–32. https://doi.org/10.7212/zkufbd.v5i1.115
13. Boutorh A, Guessoum A (2015) Classication of SNPs for breast cancer diagnosis using neural-network-based association rules. In: 2015 12th International symposium on programming and systems (ISPS). https://doi.org/10.1109/isps.2015.7244998
14. Abdel-Zaher AM, Eldeib AM (2016) Breast cancer classification using deep belief networks. Expert Syst Appl 46:139–144. https://doi.org/10.1016/j.eswa.2015.10.015
15. Karabatak M (2015) A new classifier for breast cancer detection based on Naïve Bayesian. Measurement 72:32–36. https://doi.org/10.1016/j.measurement.2015.04.028
16. Li Q, Li W, Zhang J, Xu Z (2018) An improved k-nearest neighbour method to diagnose breast cancer. The Analyst 143(12):2807–2811. https://doi.org/10.1039/c8an00189h
17. Kadam VJ, Jadhav SM, Vijayakumar K (2019) Breast cancer diagnosis using feature ensemble learning based on stacked sparse

autoencoders and softmax regression. J Med Syst 43(8):1. https://doi.org/10.1007/s10916-019-1397-z

18. Abdar M, Makarenkov V (2019) CWV-BANN-SVM ensemble learning classifier for an accurate diagnosis of breast cancer. Measurement 146:557–570. https://doi.org/10.1016/j.measurement.2019.05.022
19. Cai T (2018) Breast cancer diagnosis using imbalanced learning and ensemble method. Appl Comput Math 7(3):146. https://doi.org/10.11648/j.acm.20180703.20
20. Khuriwal N, Mishra N (2018) Breast cancer diagnosis using adaptive voting ensemble machine learning algorithm. In: 2018 IEEMA Engineer infinite conference (eTechNxT). https://doi.org/10.1109/etechnxt.2018.8385355
21. Xu H, Chen T, Lv J, Guo J (2017) A combined parallel genetic algorithm and support vector machine model for breast cancer detection. J Comput Methods Sci Eng 16(4):773–785. https://doi.org/10.3233/jcm-160690
22. Yan K, Lu H (2018) An extended genetic algorithm based gene selection framework for cancer diagnosis. In: 2018 9th International conference on information technology in medicine and education (ITME). https://doi.org/10.1109/itme.2018.00021
23. Sawhney R, Mathur P, Shankar R (2018) A firefly algorithm based wrapper-penalty feature selection method for cancer diagnosis. Lect Notes Comput Sci. https://doi.org/10.1007/978-3-319-95162-1_30
24. Sheikhpour R, Sarram MA, Sheikhpour R (2016) Particle swarm optimization for bandwidth determination and feature selection of kernel density estimation based classifiers in diagnosis of breast cancer. Appl Soft Comput 40:113–131. https://doi.org/10.1016/j.asoc.2015.10.005
25. Ahmadi A, Afshar P (2015) Intelligent breast cancer recognition using particle swarm optimization and support vector machines. J Exp Theor Artif Intell 28(6):1021–1034. https://doi.org/10.1080/0952813x.2015.1055828
26. Fallahzadeh O, Dehghani-Bidgoli Z, Assarian M (2018) Raman spectral feature selection using ant colony optimization for breast cancer diagnosis. Lasers Med Sci 33(8):1799–1806. https://doi.org/10.1007/s10103-018-2544-3
27. Yang XS, Gandomi AH (2012) Bat algorithm: a novel approach for global engineering optimization. Eng Comput. https://doi.org/10.1108/02644401211235834
28. Mirjalili S, Lewis A (2016) The whale optimization algorithm. Adv Eng Softw 95:51–67. https://doi.org/10.1016/j.advengsoft.2016.01.008
29. Neshat M, Sepidnam G, Sargolzaei M, Toosi AN (2014) Artificial fish swarm algorithm: a survey of the state-of-the-art, hybridization, combinatorial and indicative applications. Artif Intell Rev 42(4):965–997. https://doi.org/10.1007/s10462-012-9342-2
30. Mirjalili S, Mirjalili SM, Lewis A (2014) Grey wolf optimizer. Adv Eng Softw 69:46–61. https://doi.org/10.1016/j.advengsoft.2013.12.007
31. Nilashi M, Ibrahim O, Ahmadi H, Shahmoradi L (2017) A knowledge-based system for breast cancer classification using fuzzy logic method. Telematics Inform 34(4):133–144. https://doi.org/10.1016/j.tele.2017.01.007
32. Chen H-L, Yang B, Wang G, Wang S-J, Liu J, Liu D-Y (2011) Support vector machine based diagnostic system for breast cancer using swarm intelligence. J Med Syst 36(4):2505–2519. https://doi.org/10.1007/s10916-011-9723-0
33. Chauhan N, Ravi V, Karthik Chandra D (2009) Differential evolution trained wavelet neural networks: application to bankruptcy prediction in banks. Expert Syst Appl 36(4):7659–7665. https://doi.org/10.1016/j.eswa.2008.09.019
34. Jain I, Jain VK, Jain R (2018) Correlation feature selection based improved-binary particle swarm optimization for gene selection and cancer classification. Appl Soft Comput 62:203–215. https://doi.org/10.1016/j.asoc.2017.09.038
35. Naveen N, Ravi V, Rao CR, Chauhan N (2010) Differential evolution trained radial basis function network: application to bankruptcy prediction in banks. Int J Bioinspired Comput 2(3/4):222. https://doi.org/10.1504/ijbic.2010.033090
36. Friedman JH (2001) Machine. Ann Stat 29(5):1189–1232. https://doi.org/10.1214/aos/1013203451
37. Beygelzimer A, Hazan E, Kale S, Luo H (2015) Online gradient boosting. In: Advances in neural information processing systems, pp 2458–2466
38. lagus R, Lusa L (2017) Gradient boosting for high-dimensional prediction of rare events. Comput Stat Data Anal 113:19–37. https://doi.org/10.1016/j.csda.2016.07.016
39. Portugal I, Alencar P, Cowan D (2018) The use of machine learning algorithms in recommender systems: a systematic review. Expert Syst Appl 97:205–227. https://doi.org/10.1016/j.eswa.2017.12.020
40. Mirjalili S, Mirjalili SM, Hatamlou A (2015) Multi-verse optimizer: a nature-inspired algorithm for global optimization. Neural Comput Appl 27(2):495–513. https://doi.org/10.1007/s00521-015-1870-7
41. Al-Madi N, Faris H, Mirjalili S (2019) Binary multi-verse optimization algorithm for global optimization and discrete problems. Int J Mach Learn Cybern 10(12):3445–3465. https://doi.org/10.1007/s13042-019-00931-8
42. Mirjalili S, Jangir P, Mirjalili SZ, Saremi S, Trivedi IN (2017) Optimization of problems with multiple objectives using the multi-verse optimization algorithm. Knowl Based Syst 134:50–71. https://doi.org/10.1016/j.knosys.2017.07.018
43. Tegmark M (n.d.) Parallel universes. Sci Ultimate Real https://doi.org/10.1017/cbo9780511814990.024
44. Khoury J, Ovrut BA, Seiberg N, Steinhardt PJ, Turok N (2002) From big crunch to big bang. Phys Rev D. https://doi.org/10.1103/physrevd.65.086007
45. Zhu J, Ge Z, Song Z, Gao F (2018) Review and big data perspectives on robust data mining approaches for industrial process modeling with outliers and missing data. Annu Rev Control 46:107–133. https://doi.org/10.1016/j.arcontrol.2018.09.003
46. Street WN, Wolberg WH, Mangasarian OL (1993) Nuclear feature extraction for breast tumor diagnosis. Biomed Image Process Biomed Vis. https://doi.org/10.1117/12.148698
47. Wolberg WH, Mangasarian OL (1990) Multisurface method of pattern separation for medical diagnosis applied to breast cytology. Proc Natl Acad Sci 87(23):9193–9196. https://doi.org/10.1073/pnas.87.23.9193
48. Zhang J (1992) Selecting typical instances in instance-based learning. Mach Learn Proc 1992:470–479. https://doi.org/10.1016/b978-1-55860-247-2.50066-8
49. Mangasarian OL, Street WN, Wolberg WH (1995) Breast cancer diagnosis and prognosis via linear programming. Oper Res 43(4):570–577. https://doi.org/10.1287/opre.43.4.570
50. Lever J, Krzywinski M, Altman N (2016) Classification evaluation. Nat Methods 13(8):603–604. https://doi.org/10.1038/nmeth.3945
51. Kumar R, Indrayan A (2011) Receiver operating characteristic (ROC) curve for medical researchers. Indian Pediatr 48(4):277–287. https://doi.org/10.1007/s13312-011-0055-4
52. Doyle JR (1992) MCC—multiple correlation clustering. Int J Man Mach Stud 37(6):751–765. https://doi.org/10.1016/0020-7373(92)90066-t
53. Juda P, Renard P, Straubhaar J (2019) K-fold cross-validation of multiple-point statistical simulations. Pet Geostat. https://doi.org/10.3997/2214-4609.201902239
54. Levesque JC, Durand A, Gagne C, Sabourin R (2012) Multi-objective evolutionary optimization for generating

ensembles of classifiers in the ROC space. In: Proceedings of the fourteenth international conference on genetic and evolutionary computation conference—GECCO'12. https://doi.org/10.1145/2330163.2330285

55. Prasad Y, Biswas KK, Jain CK (2010) SVM classifier based feature selection using GA, ACO and PSO for siRNA design. Adv Swarm Intell. https://doi.org/10.1007/978-3-642-13498-2_40
56. Salama GI, Abdelhalim M, Zeid MAE (2012) Breast cancer diagnosis on three different datasets using multi-classifiers. Breast Cancer (WDBC) 32(569):2
57. Zheng B, Yoon SW, Lam SS (2014) Breast cancer diagnosis based on feature extraction using a hybrid of K-means and support vector machine algorithms. Expert Syst Appl 41(4):1476–1482. https://doi.org/10.1016/j.eswa.2013.08.044
58. Liu N, Qi E-S, Xu M, Gao B, Liu G-Q (2019) A novel intelligent classification model for breast cancer diagnosis. Inf Process Manage 56(3):609–623. https://doi.org/10.1016/j.ipm.2018.10.014

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.