International Journal of Mathematical, Engineering and Management Sciences
Vol. 11, No. 2, 804-830, 2026
https://doi.org/10.33889/IJMEMS.2026.11.2.034

# Random Forest Classifier with Binary Grey Wolf Optimization Feature Selection for Predicting Diabetes Mellitus

**Syaiful Anam**
Mathematics Department,
Brawijaya University, Malang, East Java, Indonesia.
*Corresponding author*: syaiful@ub.ac.id

**Dian Eka Ratnawati**
Informatics Engineering Department,
Brawijaya University, Malang, East Java, Indonesia.
E-mail: dian_ilkom@ub.ac.id

**Satrio Hadi Wijoyo**
Information System Department,
Brawijaya University, Malang, East Java, Indonesia.
E-mail: satriohadi@ub.ac.id

**Nathanael Jeshua Paat**
Mathematics Department,
Brawijaya University, Malang, East Java, Indonesia.
E-mail: nathanaeljeshua@gmail.com

(Received on December 29, 2024; Revised on May 27, 2025 & November 29, 2025; Accepted on December 27, 2025)

**Abstract**
One of the greatest challenges in medicine is the early prediction of Diabetes Mellitus (DM). This difficulty is due to the numerous clinical factors that influence prediction and the limited generalizability of data models in predicting. This issue is addressed in this study, which proposes a data prediction framework that utilizes a Random Forest Classifier and Binary Grey Wolf Optimization–based Feature Selection (RFC with BGWO-FS) along with hyperparameter tuning. This study includes extensive validation of the proposed methodology using two independent datasets with distinct clinical features. This work provided a full test of the model's stability, generalizability, and clinical applicability. Within both datasets, the proposed RFC with BGWO-FS achieved test accuracies of 78.2% and 78.8% and F1 scores of 75.3% and 70.0%, which are the highest from any other methods. The paired $t$-tests further confirm the significance of the computational time reductions over RFC with GA-FS and RFC with BPSO-FS. There is also no difference in the stability of the output from differing RFC with BGWO-FS configurations. Of the approaches analyzed, Grid Search had the highest stability in terms of generalization, while Bayesian Optimization did so with the least computation overhead. The clinically relevant features of general health, hypertension, cholesterol, glucose, and BMI selected by SHAP confirmed the relevance of the features. The results also demonstrated that across both datasets, RFC with BGWO-FS is accurate and generalizable. The results of this analysis should support the integration of RFC and BGWO-FS into diabetes screening workflows and clinical decision support systems.

**Keywords-** Diabetes mellitus prediction, Feature selection, Random forest classifier, Grey wolf optimization.

## 1. Introduction

Diabetes Mellitus (DM) is defined as a disorder of metabolism that results in difficulty with one's body's ability to control glucose levels in the bloodstream. Indonesia is now ranked fifth in the world with DM (Prakoso et al., 2023) while all over the world cases of the disorder are expected to rise from 537 million to more than 783 million from 2021 to 2045 (Burgess et al., 2024). The disorder is linked to a vast range of

severe complications like cardiac illnesses, neuropathy, retinopathy, kidney failure, and sexual dysfunction (Papatheodorou et al., 2015) while in Indonesia, it is even responsible for 6.7% of mortality rate (Marlina et al., 2024). Given its rising public health concern and the impact on the population, the early detection of the disorder becomes quite imperative. Contrary to that though is the fact that current immeasurable techniques in detection of the disorder rely on the input of a specialized medical personnel and result in very expensive, inaccessible, and at times inconsistent, medical diagnosis. This is the reason we are in need of unconventional ways of screening to provide measurable, automated, and cost-effective diagnosis for the DM in the population.

The increasing complexities in cases of DM as well as in untangling healthcare costs have opened up the possibility of Detecting DM using Machine Learning (ML) (Predictable Model) (Obermeyer & Emanuel, 2016). Predictive healthcare models more generally have the potential to minimize the risk of responding too late in treatment (Predictable model) (Olchanski et al., 2024). The Spectrum of ML models such as (1) Algorithms using Self-Organizing Maps (SOM), (2) Tree Algorithms (3) Ensemble (4) Support-vector Machine (SVM), (5) Logistic Regression, and Advanced Ensembles, have resulted in and continue to record a high-performance rating in the determination of the DM Class (Zou et al., 2018; Mohan et al., 2019; Alzboon et al., 2023). However, among the above models, the Random Forest Classifier (RFC) has proven to be the most superior in versatility, uniformity in the rate of prediction as well as in the robustness of the model (Rodriguez-Galiano et al., 2012; Shatnawi et al., 2014; Masud et al., 2024), thus it has been used more in the medical field, as well as in the industry (Xie et al., 2009; Yego et al., 2021; Minnoora & Bathsb, 2023; Nguyen et al., 2023; Mahmood & Abdullah, 2024).

RFC is affected by the input feature relevance and the performance issues. Inaccuracy, increased computational intensity, and weaker interpretability stem from the presence of irrelevant or redundant attributes (Tama & Rhee, 2018; Louk & Tama, 2022). All feature selection techniques (filter, wrapper, and embedded) can alleviate the issues of irrelevance and redundancy, although wrapper methods are generally superior with regard to the feature interaction in prediction methods (Kordon, 2010). In machine learning studies focusing on the prediction of diabetes mellitus, appropriate feature selection has been shown to enhance classification performance (Sheta et al., 2024). In this context, Swarm Intelligence (SI) algorithms are being applied to feature selection, and more specifically, to high-dimensional data (Brezočnik et al., 2018). There is no shortage of methods to address high dimensionality with SI algorithms, including Particle Swarm Optimization (PSO), Genetic Algorithm (GA), Bat Algorithm (BA), and Grey Wolf Optimization (GWO) (Ansyari et al., 2023; Kangra & Singh, 2024).

GWO was developed by Mirjalili et al. (2014). GWO has become a well-known nature-inspired optimizer (Behera & Mohanty, 2019; Liu et al., 2021), because of its balanced exploration-exploitation mechanism and remarkable global search capabilities. Research has found GWO to be better than PSO, BA, and GA in a number of optimization problems (Nimma et al., 2018; Kong & Ma, 2018). However, several hybrid and enhanced versions of the basic GWO have been developed to address its shortcomings, especially regarding convergence and optimization performance (Zhang et al., 2017). More recently, the improvement of GWO in the medical field has been the focus of research, where GWO was used and modified to classify diabetes (Sam'an et al., 2025), and a new feature addition based on autophagy GWO was used to improve feature selection (Sirmayanti et al., 2025). Improvements in the same span of time in Teaching-Learning Based Optimization (TLBO) are also considerable in the fields of multi-response optimization and wrapper-based feature selection (Ang et al., 2022; Pan et al., 2025). Furthermore, metaheuristic optimization of deep learning in various domains has shown noticeable improvement. One of these domains is wafer defect classification (Ang et al., 2023).

Nevertheless, available studies on feature selection using GWO or TLBO focus on one dataset. They are statistically unsound and do not consider model interpretability. It harms clinical authenticity and generalizability. Furthermore, there has been limited research combining Binary GWO feature selection with RFC (RFC with BGWO-FS) and its evaluation across diverse DM datasets. Driven by these shortcomings, this research aims to incorporate Binary GWO to reduce dimension problems along with RFC for boosted DM classification. This research aims to achieve the following goals:

a. boost the efficiency of RFC by providing optimal feature subset;
b. help provide a cost efficient and more reliable alternative for DM classification; and
c. help lessen the amount of manual work needed by automating the feature selection process to optimize RFC.

## 2. Method

This paper hybridizes the RFC and BGWO-FS to construct a new method for diagnosing DM. The principal activities of the proposed methodology are depicted by the flowchart in **Figure 1**. The activities of the proposed methodology include data preprocessing, partitioning and transformation of the dataset, model building using RFC integrated with BGWO-FS, setting input data and GWO parameters, and evaluating the model. The purpose of this paper is to construct a well-performing and reliable classification model for DM and thoroughly investigate the model's generalization ability on different datasets.

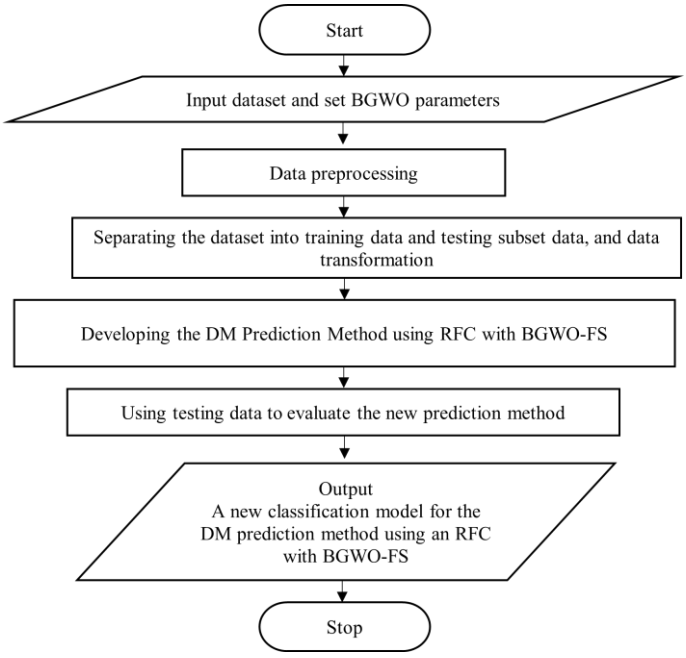



**Figure 1.** The classification model's flowchart for the prediction method using the RFC with BGWO-FS.

### 2.1 Dataset and Configuration of BGWO Parameters

For the purpose of evaluating the proposed method in this research, two datasets are used. The 1st dataset is titled "diabetes_binary_health_indicators_BRFSS2015.csv" which is available from Kaggle and contains 253,680 data records from a large-scale health survey. The target variable in this data is binary ($D$) i.e. diabetes/prediabetes (1) vs. no diabetes (0). The dataset comprises of 21 attributes such as physical health, health-related behaviours, health care access, and sociodemographic characteristics. A description of the

attributes is provided in **Table 1**. The 2nd dataset used in this research is Pima Indian Diabetes Database which is also available on Kaggle. The dataset has 8 attributes and a total of 768 data records.

The variables in the dataset include demographic information, measurements of the body, biochemical test results, data from physical examination records, family medical history, in addition to the target variable (Outcome). The description of the features of the 2nd dataset can be seen in **Table 2**. Due to the class distribution being heavily weighted towards the majority class, appropriate evaluation metrics are required. The use of two separate datasets, each with its own set of features and differing distributions allows for a more thorough test of the external validity and robustness of the approach proposed. This so called dual-dataset approach mitigates the chances of dataset specific bias while providing the opportunity to evaluate if the developed model has the ability to generalize to other settings for the population. Feature selection done using BGWO, with multiple hyperparameter settings to be tuned. This study employs the use of different population sizes for the wolves of 5, 10, and 20, while a maximum of 1000 iterations ($t_{\max}$) was set. Furthermore, an early stopping condition was implemented: the procedure would stop if the global best solution was unchanged for 100 iterations, a sign of stagnation with no improvement. The fitness function was defined as 1 minus the F1-score on the validation subset, resulting in the search for feature subsets that would yield a more balanced solution in class distribution, rather than simply increasing the accuracy.

## 2.2 Data Pre-processing

Data preprocessing is the first step in the data mining and ML workflow. Its objective is to improve the data quality. It also ensures that the analytical models are able to retrieve relevant insights. It involves data-harmonization techniques for raw data as well as removing irrelevant and duplicate data, or data that may exhibit hostility to the analytical techniques. It also addresses some common challenges/problems in data mining, such as missing values, faulty or inconsistent formatting, and noise. In this research, missing numerical values were addressed by median imputation, which is a technique that lessens the influence of outliers, while the missing categorical values were imputed by using the mode. One-hot encoding was used to transform categorical variables to ensure that the features are aligned with the requirements of the RFC.

Since the dataset is relatively balanced, with class 1 comprising 48.9% and class 0 comprising 51.1%, data augmentation or resampling techniques such as oversampling or under sampling were not applied. Therefore, we don't use data augmentation or resampling techniques such as oversampling or under sampling. This decision was made to avoid introducing synthetic bias and to preserve the natural distribution of the data. The Pima dataset is more imbalanced. We relied on F1-score and recall as primary metrics, rather than applying resampling, in order to objectively evaluate the classifier's ability to handle class imbalance without artificially modifying the data distribution.

**Table 1.** Features description of the 1st dataset.

| Features | Descriptions |
|---|---|
| *HBP* | High Blood Pressure ($HBP$) indicates high blood pressure: '0' means normal, '1' means high. |
| *HC* | Binary indicator of high cholesterol ($HC$): '0' = no, '1' = yes. |
| *NoChol* | '1' if the patient had a cholesterol test in the last 5 years, otherwise '0'. |
| *BMI* | Body Mass Index (BMI) value. |
| *Smkr* | '1' if the person has smoked at least 100 cigarettes (*Smoker*); '0' otherwise. |
| *Strk* | Stroke ($Strk$) status: '0' = no stroke, '1' = stroke occurred. |
| *HDA* | '1' if the person had a heart attack (MI) or coronary heart disease (CHD); '0' if not. |
| *FA* | Physical activity ($FA$) in the past month (excluding work): '1' = active, '0' = inactive. |
| *F* | Fruit ($F$) consumption: '1' = eats fruit daily, '0' = does not. |
| *Vg* | Vegetable ($Vg$) consumption: '1' = eats vegetables daily, '0' = does not. |
| *HD* | Heavy drinking ($HD$) indicator: '1' = yes, '0' = no (based on gender-specific thresholds). |
| *Hc* | Health coverage ($HC$) status: '1' = covered (e.g., insurance, HMO), '0' = not covered. |
| *NDC* | '1' if the person couldn't afford to see a doctor in the past year; '0' = no issue. $NDC$ (Needed Doctor but Couldn't). |

Table 1 continued…

| | |
|---|---|
| *GH* | General health (*GH*) rating: 1 = excellent to 5 = poor. |
| *MH* | Mental health is represented by $MH$. The number of days in the last 30 days that were mentally unwell (1–30 scale). |
| *PH* | $PH$ defines physically healthy. The number of physically unwell days in the previous 30 days as a result of sickness or injury is represented by $PH$. |
| *DW* | Difficulty walking (*DW*) or climbing stairs: '1' = yes, '0' = no. |
| *Sex* | '0' = female, '1' = male. |
| *Age* | Age category (1–13), e.g., 1 = 18–24, 9 = 60–64, 13 = 80+. |
| *ED* | Education level (1–6): 1 = no school/kindergarten, 6 = college graduate. |
| *Incm* | Income bracket (1–8): 1 = <$10,000, 5 = <$35,000, 8 = ≥$75,000. |
| *D* | Diabetes (*D*) status: '1' = has diabetes, '0' = does not. |

**Table 2.** Features description of the 2nd dataset.

| **Features** | **Descriptions** |
|---|---|
| Pregnancies | Number of times the patient has been pregnant. |
| Glucose | Plasma glucose concentration measured 2 hours after an oral glucose tolerance test. |
| BloodPressure | Diastolic blood pressure (mm Hg). |
| SkinThickness | Triceps skin fold thickness (mm), indicating subcutaneous fat. |
| Insulin | Serum insulin level (μU/ml). |
| BMI | Body Mass Index, a measure of body fat calculated as weight (kg) / height (m²). |
| DiabetesPedigreeFunction | Score indicating genetic risk of diabetes based on family history (pedigree function). |
| Age | Age of the patient in years. |
| Outcome | Diabetes diagnosis: '1' = diabetic, '0' = non-diabetic. |

## 2.3 Partitioning the Dataset into Testing and Training Subset, and Data Transformation

An important part of creating a strong ML or data-mining model is splitting the data set into training and test subsets. Generally, a training subset consists of 70-80% of the total data. A model is trained and parameter values are set using the training subset. The remaining data is put into the test subset. A model is evaluated on new data, then the performance of the model is set using the test subset. One may assess a model to determine the extent to which a model can generalize based on the performance of the model on the test subset. A rational way of validating performances is provided by separating the data. It also uncovers issues such as underfitting (poor performance on both the training and test sets) and overfitting (high performance on the training set but poor performance on new data). For this study, 30% of the data was set aside for testing and 70% was used for training. The split data was done using stratified sampling to maintain the same proportions of classes in both the training and testing subsets. In experiments which require a validation subset (like BGWO fitness evaluation and hyperparameter tuning), the training data is split even further into internal training and validation folds. Prior to modelling, data normalization needs to be done.

An important preprocessing step is normalization, which affects the common scale of feature values. Normalization of data is important for distance-based and gradient-based techniques. Moreover, feature values that are large numerically should not dominate the learning of the model. Normalization of the data can improve the model training by speeding convergence, reducing model training time, and improving accuracy model. Increased model interpretability and reliability result from normalization of the data. This step is done after splitting the data in order to avoid the leakage problem. In this work, all the continuous features were normalized using Min-Max scaling to the [0, 1] interval, with the scaling parameters being the minimum and the maximum, which were calculated only on the training subset and applied to the validation and the test subsets. This procedure is designed to avoid data leakage and to make sure that the test set does not leak into the model training or feature selection.

## 2.4 Develop a DM Prediction Method using RFC with BGWO-FS

This subsection concerns the work that has gone into implementing DM prediction using an RFC method alongside BGWO-FS. RFC methods were first proposed by Breiman and Cutler. RFC are among some of the most robust algorithms and fall under the category of ensemble algorithms designed for classification tasks. However, before working with this RFC, it is crucial that we appreciate the RFC methodology. During the training stage, the RFC methods generate multiple decision trees, and each one votes on the prediction, which is aggregated using some voting mechanism. For classification tasks, the RFC methodology produces a tentative prediction, and the class with the most votes is selected.

RFC method combines the outputs of diverse decision trees built on random feature subsets. By using this strategy, the herring of trees creates a strong ensemble, decreases the probability of overfitting, and boosts predictive performance. **Figure 2** showcases RFC methodology designed for classification problems, and each tree votes for classification, and the votes are aggregated using an ensemble approach. These schematics portray that the methodology improves the generalization of the model. Consequently, RFC has the most relevance to high dimensional and large volume data sets like the data sets in the medical fields, which is why it is particularly relevant in building diagnosis predictive systems like the one in this work.

Assume that there are $p$ predictor variables and $n$ observations in the training set of data. Here is a straightforward statement of the RFC algorithm: (Breiman, 2001).

1) Bootstrap Stage
   Draw random samples of size *n* from the training data with replacement.
2) Stage of Random Sub Setting
   Using the bootstrap dataset, build a decision tree to its maximum size without pruning. At each node, the splitting variable is selected by randomly choosing *m* predictor variables, where $m < p$, and then choosing the best split among these *m* variables.
3) Repeat steps 1 and 2 for *k* iterations to generate a forest consisting of *k* decision trees.
4) Voting Stage
   For each prediction, use majority voting in classification problems (i.e., choose the mode), and use the average in regression problems (i.e., compute the mean).
5) The final prediction is the one that appears most frequently across all trees in the forest.

In this study, the features for the input of the RFC are chosen by using GWO. RFC performance will be enhanced by the specific characteristics. **Figure 1** shows the flowchart for the DM Prediction Method utilizing RFC with BGWO-FS. The suggested approach consists of many phases. In the initial step, the dataset is entered and divided into training and testing subset data. Additionally, other BGWO parameters, such as the number of wolves ($f$) and the maximum number of iterations ($t_{\max}$), should be specified.

Each wolf's position is initialized with random numbers. The location of each wolf determines the selected RF features. The specified location is determined by Equation (1).

$$\boldsymbol{X}_i(t) = \left(X_{i,1}(t), X_{i,2}(t), \ldots, X_{i,\boldsymbol{Nfeat}}(t)\right), i = 1, \ldots, N_{\text{wolf}} \quad (1)$$

The wolf population is denoted by $N_{\text{wolf}}$. The $i^{\text{th}}$ wolf presents a candidate feature subset for RFC. It is denoted by $\boldsymbol{X}_i(t)$. $t$ stands for iteration. Either 1 or 0 is the value of $X_{\text{i,j}}(t)$. The feature that is utilized is represented by the number 1, while the feature that is not used is represented by the value 0.

Using Equation (2), the fitness function value is then determined. Using a classification model with RFC and features derived from $\boldsymbol{X}_i(t)$, the fitness function is 1- F1 score. In Pseudocode 2, the fitness function pseudocode is displayed.

$$f_i = fitness(\boldsymbol{X}_i(t), \text{X\_train}, \text{y\_train}, \text{X\_test}, \text{y\_test}) \quad (2)$$

The next step is to find $\boldsymbol{X}_\alpha(t)$, $\boldsymbol{X}_\beta(t)$, and $\boldsymbol{X}_\delta(t)$. The initial best solution with the lowest fitness value is $\boldsymbol{X}_\alpha(t)$, followed by $\boldsymbol{X}_\beta(t)$, which has the second-smallest fitness value, and $\boldsymbol{X}_\delta(t)$, which has the third-smallest fitness value.

Equations (3) and (4) illustrate the typical shape of grey wolf activity while encircling prey.

$$\boldsymbol{D}(t) = \left|\boldsymbol{C}(t) \odot \boldsymbol{X_p}(t) - \boldsymbol{X_i}(t)\right| \quad (3)$$

$$\boldsymbol{X_i}(t+1) = \boldsymbol{X_p}(t) - \boldsymbol{A}(t) \odot \boldsymbol{D}(t) \quad (4)$$

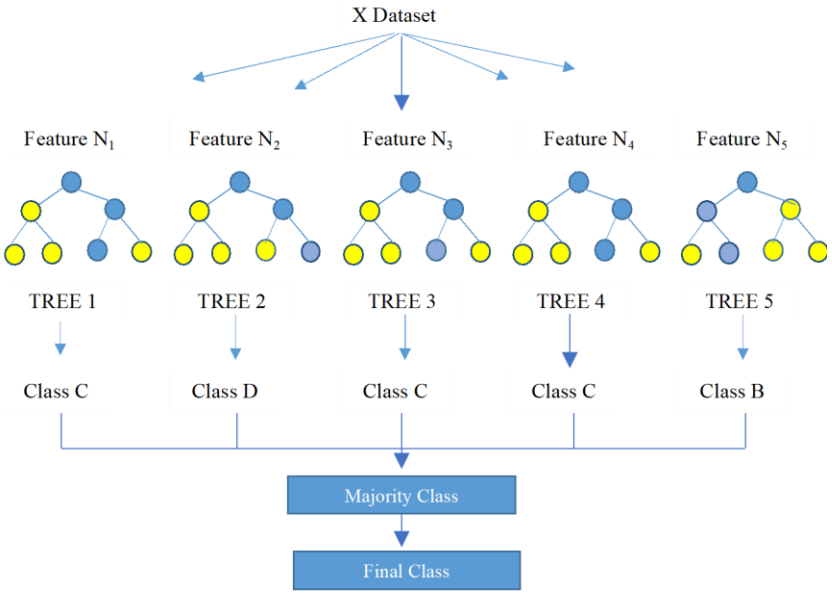



**Figure 2.** Illustration of RFC.

The $i^{\text{th}}$ grey wolf's position at time $t$ is denoted by $\boldsymbol{X_i}(t)$ and $\boldsymbol{X}_p(t)$ is where the prey is located. The distance between the wolf's and prey's positions is represented by the vector $\boldsymbol{D}(t)$. The wolf's position in the next time step is denoted by $\boldsymbol{X_i}(t+1)$. $\boldsymbol{A}$ and $\boldsymbol{C}$ are vector coefficients. The Hadamard Product operation is denoted by the symbol $\odot$. Equations (5) and (6), where an is a vector, whose value drops linearly from 2 to 0 during the iteration process, may be used to derive parameters $\boldsymbol{A}$ and $\boldsymbol{C}$.

$$\boldsymbol{A}(t) = 2\boldsymbol{a}(t) \odot \boldsymbol{r}_1 - \boldsymbol{a}(t) \quad (5)$$

$$\boldsymbol{C}(t) = 2\boldsymbol{r}_2 \quad (6)$$

Random values from the interval [0,1] make up the vectors $\boldsymbol{r_1}$ and $\boldsymbol{r_2}$. Equation (7), where $t_{\max}$ is the maximum number of time steps, may be used to update the value of $\boldsymbol{a}$.

$$\boldsymbol{a}(t) = \mathbf{2} - \mathbf{2}\left(\frac{t}{t_{\max}}\right) \quad (7)$$

Hunting serves as the second mechanism. The wolf will modify its location according to the alpha, beta, and delta wolves' placements. It is difficult to determine the exact position of the wolf, so the distance between the $i^{\text{th}}$ wolf and the predator is estimated using the three best wolves as expressed by Equations (8) and (9). The vectors $\boldsymbol{D_a}$, $\boldsymbol{D_\beta}$, and $\boldsymbol{D_\delta}$ represent the range of positions between positions of wolves and the best positions of alpha, beta, and delta wolves, respectively, as an approximation of prey position.

$$\begin{aligned} \boldsymbol{D_a}(t) &= |\boldsymbol{C_1}(t) \odot \boldsymbol{X_\alpha}(t) - \boldsymbol{X_i}(t)| \\ \boldsymbol{D_\beta}(t) &= |\boldsymbol{C_2}(t) \odot \boldsymbol{X_\beta}(t) - \boldsymbol{X_i}(t)| \\ \boldsymbol{D_\delta}(t) &= |\boldsymbol{C_3}(t) \odot \boldsymbol{X_\delta}(t) - \boldsymbol{X_i}(t)| \end{aligned} \tag{8}$$

$\boldsymbol{C_1}, \boldsymbol{C_2}$, and $\boldsymbol{C_3}$ in Equation (8) can be calculated by using Equation (6).

$$\begin{aligned} \boldsymbol{Bstep}_1 &= \frac{1}{1+e^{-10\boldsymbol{A}_1 \odot \boldsymbol{D}_\alpha}} \\ \boldsymbol{Bstep_2} &= \frac{1}{1+e^{-10\boldsymbol{A}_2 \odot \boldsymbol{D}_\beta}} \\ \boldsymbol{Bstep_3} &= \frac{1}{1+e^{-10\boldsymbol{A}_3 \odot \boldsymbol{D}_\delta}} \end{aligned} \tag{9}$$

The parameter values $\boldsymbol{A_1}, \boldsymbol{A_2}$, and $\boldsymbol{A_3}$ may be found by using Equation (5).

$$\begin{aligned} X_{1,j} &= \begin{cases} 1, X_{\alpha,j} + Bstep_{1,j} \geq 1 \\ 0, X_{\alpha,j} + Bstep_{1,j} < 1 \end{cases}, j = 1, \ldots, N_{\text{feat}} \\ X_{2,j} &= \begin{cases} 1, X_{\beta,j} + Bstep_{2,j} \geq 1 \\ 0, X_{\beta,j} + Bstep_{2,j} < 1 \end{cases}, j = 1, \ldots, N_{\text{feat}} \\ X_{3,j} &= \begin{cases} 1, X_{\boldsymbol{\delta},j} + Bstep_{3,j} \geq 1 \\ 0, X_{\boldsymbol{\delta},j} + Bstep_{3,j} < 1 \end{cases}, j = 1, \ldots, N_{\text{feat}} \end{aligned} \tag{10}$$

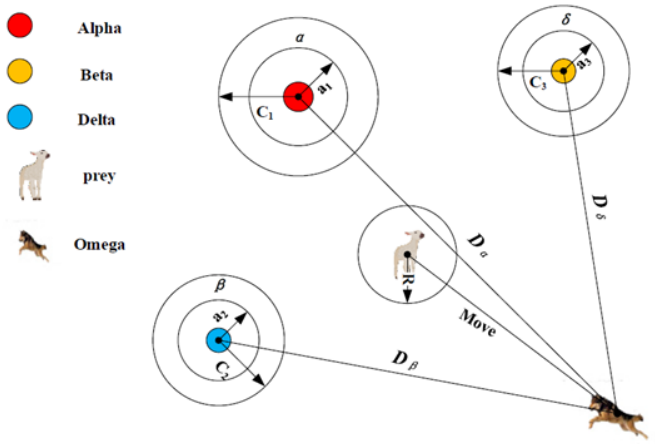



**Figure 3.** Wolf position update (Zhao et al., 2019).

Equation (10) that approximates the prey position will then be used to update all wolf or agent positions. Equation (11) is used to get the $i^{\text{th}}$ wolf's location update at time $t + 1$. $r$ is a uniformly distributed random between 0 and 1.

$$X_{i,j}(t+1) = \begin{cases} X_{1,j}, r < 1/3 \\ X_{2,j}, 1/3 \le r < 2/3 \\ X_{3,j}, r \ge 2/3 \end{cases}, i = 1, \dots, N_{\text{wolf}};\ j = 1, \dots, N_{\text{feat}} \quad (11)$$

Equation (11), which approaches the prey's position, will then be used to update all wolf or agent positions. **Figure 3** provides an example of the position update.

***Pseudocode* 1.** Random Forest Classifier (RFC) integrated with Binary Grey Wolf Optimization–based Feature Selection (BGWO-FS)
Initialize the population of grey wolves $N_{\text{wolf}}$
Initialize the objective function $O$ and dimension dim
Initialize the maximum iteration $t_{\max}$
Initialize $t = 0$, $L = 0$.
***for*** *i=1 to* $N_{\text{wolf}}$ ***do***
$\boldsymbol{X_i}$ **= int** (*random* ($N_{\text{wolf}}$, $N_{\text{feat}}$))
***end for***
***for*** *i=1 to* $N_{\text{wolf}}$ ***do***
Determine each wolf agent's fitness value using the objective function.
$fitness_i = O(\boldsymbol{X}_i)$
***end for***
Calculate $\boldsymbol{X}_\alpha$, $\boldsymbol{X}_\beta$, and $\boldsymbol{X}_\delta$
Calculate fitness of $\boldsymbol{X}_\alpha$, $\boldsymbol{X}_\beta$, and $\boldsymbol{X}_\delta$
***while*** $t < t_{\max}$ and $L <= 100$ ***do***
$\boldsymbol{a} = \mathbf{2} - \mathbf{2}\left(\frac{t}{t_{\max}}\right)$
***for*** *i=1 to* $N_{\text{wolf}}$ ***do*** (for every wolf agent)
***for*** *j=1 to* $N_{\text{feat}}$ ***do***
***for*** *q=1 to* 3 ***do***
Initialize arbitrary $\boldsymbol{r}_1$ and $\boldsymbol{r}_2$
Calculate the values of $\boldsymbol{A}_q$ and $\boldsymbol{C}_q$
$A_q[j] = 2a[j]r_1[j] - a[j]$
$C_q[j] = 2r_2[j]$
*end* ***for***
*Update* agent position
$D_\alpha[j] = |C_1[j]\, X_\alpha[j] - X_i[j]|$
$D_\beta[j] = |C_2[d]\, X_\beta[j] - X_i[j]|$
$D_\delta[j] = |C_3[d] X_\delta[j] - X_i[j]|$

$$Bstep_1[j] = \frac{1}{1+e^{-10A_1[j]D_\alpha[j]}}$$
$$Bstep_2[j] = \frac{1}{1+e^{-10A_2[j]D_\beta[j]}}$$
$$Bstep_3[j] = \frac{1}{1+e^{-10A_3[d]D_\delta[j]}}$$

$$X_1[j] = \begin{cases} 1, X_\alpha[j] + Bstep_1[j] \ge 1 \\ 0, X_\alpha[j] + Bstep_1[j] < 1 \end{cases}$$

$$X_2[j] = \begin{cases} 1, X_\beta[j] + Bstep_2[j] \geq 1 \\ 0, X_\beta[j] + Bstep_2[j] < 1 \end{cases}$$
$$X_3[j] = \begin{cases} 1, X_\delta[j] + Bstep_3[j] \geq 1 \\ 0, X_\delta[j] + Bstep_3[j] < 1 \end{cases}$$

$r = random()$
**if** $r < 1/3$ **then**
$X_i\,[j] = X_1[j]$
**else if** $r < 2/3$ **then**
$X_i\,[j] = X_2[j]$
**else**
$X_i\,[j] = X_3[j]$
**end if**
***end for***
***end for***
***for*** *i=1 to* $N_{wolf}$ ***do***
Calculate fitness value and update new position of each agent
***if*** $fitness_i$ new < $fitness_i$ old **then**
$\boldsymbol{X}_i$ = $\boldsymbol{X}_i$ new
***else***
$\boldsymbol{X}_i = \boldsymbol{X}_i$ old
*end* ***if***
***end for***
*Update* $\boldsymbol{X}_\alpha$, $\boldsymbol{X}_\beta$, dan $\boldsymbol{X}_\delta$ by considering the new position's fitness worth -$t^{th}$
**if** the fitness of $\boldsymbol{X}_\alpha$ doesn't change **then**
$L = L + 1$
**else**
$L = 0$
**end**
$t = t + 1$
***end while***
***output*** $\boldsymbol{X}_\alpha$

***Pseudocode* 2.** Fitness function
Function fitness($\boldsymbol{x}, \boldsymbol{X\_train}, \boldsymbol{y\_train}, \boldsymbol{X\_test}, \boldsymbol{y\_test}$)
a. Take features $\boldsymbol{X\_train}$ and $\boldsymbol{X\_test}$ based $\boldsymbol{x}$,
b. rf = RandomForestClassifier()
c. rf.fit($\boldsymbol{X\_train}, \boldsymbol{y\_train}$)
d. $\boldsymbol{y\_pred}$ = rf_classifier.predict($\boldsymbol{X\_test}$)
e. $f_1$ = $f_1$_*score*($\boldsymbol{y\_test}, \boldsymbol{y\_pred}$)
f. fit=1-$f_1$
g. return fit

The execution of the proposed method terminates once specific stopping criteria are satisfied. These include completing the maximum number of repeats or not seeing the fitness value increase any further. The algorithm assumes that the global optimum has been achieved if there is no enhancement in the best global fitness over 100 consecutive iterations. The BGWO is employed to select relevant features for the

classification models. The optimization process ends and its outcome is stored if the termination criterion is satisfied. This output consists of the selected feature subset to be used by the RFC.

**Algorithm 1.** Assessment of a DM Prediction Method by using the RFC with BGWO-FS.
**Input:**
The training data (***X_train***) with size of $n \times m$
Features obtained by BGWO
***y_train*** (Training data set's class label)
**Output:**
Accuracy, recall, precision, F1 score.
1) Train RF Model using training data with features obtained by BGWO.
2) Determine the label prediction ***y_pred*** based on RFC.
3) Determine accuracy, recall, precision and F1 score.

**Algorithm 2.** Evaluation of the DM Prediction Method using RFC with BGWO-FS.
**Input:**
The testing data with features obtained by BGWO
RF model
***y_test*** (each testing data's class labels)
**Output:**
Accuracy, recall, precision, F1 score.
1) Determine the label prediction ***y_pred*** based on RFC.
2) Determine accuracy, recall, precision and F1 score.

## 2.4 Hyperparameter Optimization and Model Evaluation

Three hyperparameter optimization methods were used to improve the predictive performance of the model and include Grid Search, Random Search, and Bayesian Optimization. These methods allow for a more robust and balanced exploration of the hyperparameter space of the RFC using deterministic, stochastic and model-based search approaches.

***Grid Search Optimization***

A structured hyperparameter search was set up, focusing on the parameters of the RFC that are known to impact, to a large extent, the depth, complexity and generalization of the model. The following ranges were investigated: $bootstrap$ in $\{True\}$, $max\_depth$ in $\{10, 50, 80, 100\}$, $max\_features$ in $\{\text{“}sqrt\text{”}\}$, $min\_samples\_leaf$ in {3, 4, 5}, min_samples_split in {8, 10, 12} and $n\_estimators$ in $\{100, 200, 300, 1000\}$. This setup led to the RFC hyperparameter space containing 144 candidate combinations, making it possible to perform a fully exhaustive search that is still computationally feasible.

***Random Search Optimization***

This method of search also used the same parameter ranges, but Random Search Optimization did not consider all candidate combinations, and instead it sampled 100 random combinations. At a lower computational cost, this stochastic method allows for a broader exploration of the hyperparameter space and is useful when only a few parameters are of high importance to the model.

***Bayesian Optimization***

With Bayesian Optimization, a surrogate model (Gaussian Process) was used along with Expected Improvement (EI). This technique evenly distributes adaptive exploration and exploitation by estimating where across the hyperparameter the maximum potential performance could lie, and focusing on the most

probable contenders. Bayesian Optimization provides similarly effective performance as the exhaustive search while utilizing considerably fewer evaluations of the model.

***Model Evaluation***

After hyperparameter tuning, the RFC model was trained on the feature subset BGWO-FS identified. This model was then ascertained for its generalization capability on an independent test dataset. Evaluation was done on both training and test subset using the quartet of standard classification metrics; these being accuracy, precision, recall, and F1 score. The values of both the training and test datasets are computed.

a. Another popular name for the categorization rate is accuracy. The formula given in Equation (12) is used to compute it. In relation to the total number of instances in the dataset, it measures the proportion of accurately predicted cases. The predictive performance of the model is generally shown by this formula.

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (12)$$

The number of cases that the model properly identifies as positive instances is known as a True Positive ($TP$). When the model correctly predicts a negative occurrence, it is referred to as a True Negative ($TN$). However, a Type I mistake is another name for a False Positive ($FP$). It occurs when a negative occurrence is inadvertently classified as positive by the model. On the other hand, when a positive occurrence is mistakenly forecasted as negative, it is known as a False Negative ($FN$) or Type II error.

b. It calculates the percentage of real positive examples that the model accurately detects. Recall indicates how well an algorithm can find the actual positives. Missing the detection of actual positives can lead to severe consequences.

$$Recall = \frac{TP}{TP+FN} \quad (13)$$

c. As stated in Equation (14), precision is defined as the ratio of correctly predicted positives to the overall number of instances predicted as positives by the model. This formula is very helpful when determining the effectiveness of the model's positive predictions, as well as when trying to mitigate false positive predictions.

$$Precision = \frac{TP}{TP+FP} \quad (14)$$

d. In Equation (15), the F1 score is explained as the mean of accuracy and recall. It is a good measure of model performance. It helps a lot when positive class and negative class are imbalanced, especially when precision alone can be misleading.

$$F1\ score = 2 \cdot \frac{Precision \cdot Recall}{Precision+Recall} \quad (15)$$

To facilitate fairness and exhaustiveness in evaluating the requested RFC and BGWO-FS method, comparisons have been made with several other algorithms. In this case, the other algorithms, or baselines, include classic algorithms of machine learning (such as Decision Tree, KNN, SVM, Logistic Regression, Gaussian Naïve Bayes, AdaBoost, RFC, Histogram Gradient Boosting) and Deep Learning (DNN), two other algorithms that depend on heuristic optimization and feature selection (BPSO-FS and GA-FS). Having the proposed framework in conjunction with the supporting baseline models enables comparison in predictive ability and other aspects including stability and speed of the algorithm. Equally shared experimental paradigms and settings have been established to ensure fairness and uniformity in comparison.

***Statistical Validation***

Statistical validation of the differences in experimental observations was initiated with a two-tailed paired sample t-test establishing a cut-off of 5 percent ($\alpha = 0.05$). This test is aimed towards the proposed BGWO-FS-enhanced RFC, in comparison with baseline RFC, GA-FS and BPSO-FS over the set of iterative experimental runs. In this instance, the statistical appraisal of the F1 score and time taken to compute suggests the changes noted in the values was as a result of the method used.

***Model Interpretability***

With regards to Model Interpretability to ensure that the results are clinically reasonable and adequately explained, SHAP analysis was performed with TreeExplainer. Global SHAP summaries were employed to single out the diabetes risk predictors which had the greatest impact, and individualized SHAP explanations were used to rationalize the outputs of the model. This transparency improves real-world decision support systems' adaptability and integration.

## 3. Results and Discussions

Four of the most commonly used measures of classification, which are accuracy, precision, recall, and F1-score, were used for assessing the performance of the proposed process. The dataset had to go through a number of preprocessing steps to ensure reliable performance. To begin with, the data were checked for missing values, and as there were no missing values, there was no need for imputation. The next step was to perform outlier detection by using the Interquartile Range (IQR) method, as the extreme values could negatively impact the learning. To reduce the noise and to ensure the feature space was homogeneous, all outliers were removed. Data enhancement or oversampling strategies were avoided due to the presence of a relatively balanced class distribution (48.9% class 1 and 51.1% class 0); thus, maintaining the original dataset structure was appropriate.

After the preprocessing steps were completed, the dataset was stratified and split 70:30. The features were normalized by using a method called Standard Scaler. The mean and standard deviation were only calculated on the training set to avoid data leakage. These parameters that were derived from the training set were then applied to both training and testing sets. Standardization is a key aspect for improving the behaviour of an algorithm with regards to its convergence, numerical stability, and prediction accuracy, particularly for ML variants susceptible to the amount of feature magnitude.

Furthermore, due to the fact that RFC and other ML methods include some degree of randomness, such as selecting features at random and bootstrap sampling, it is essential to run the model several times in order to estimate its performance reliably. Taking the mean metric of the model performance over the multiple runs to mitigate the effects of random initialization and sampling makes the evaluation of the model much more stable. Hence, the final outputs are the mean and standard deviation over the multiple independent runs.

For 1st dataset, **Tables 3** and **4**, demonstrate that the RFC combined with BGWO-FS achieves the highest level of performance with respect to both training and testing running without exception. Each of the simulations with 5, 10, and 20 wolves achieves record training results in terms of both the training accuracy and the training F1-score of 0.996 to 0.999. The marginal gain of performance exhibited within the testing wolves beyond 10 wolves, indicates that population of 10 wolves is sufficient along with a high performance on the training record for non-basic wolf population trainers. The testing results show a balanced high performance of all tested configurations with an accuracy of 0.747 to 0.748, in F1-score to be 0.746-0.748 and a comparable precision and recall basis of 0.77 to 0.78 with respect to each to a record of 0.726.

When compared with other optimization-based methods, the RFC with BGWO-FS is distinctly superior. The RFC with BGWO-FS still outperformed the rest. All of the other classical benchmark classifiers, such as were the Gaussian NB and KNN, SVM, AdaBoost, MLP, were also shown with very low accuracy. The Decision Tree model showed exemplary training results with an accuracy of one, but it exhibited poor generalization ability, evidencing extreme overfitting. Other solid ensemble models, such as the baseline Random Forest and Histogram Gradient Boosting, performed slightly worse than BGWO-FS in the initial stages of testing. In conjunction with the wrapper-based feature selection for structured medical data, Neural Networks, such as MLP and DNN, underperformed in achieving the balanced precision–recall score relative to BGWO-FS.

**Table 3.** Comparison of training accuracy, precision, recall, and F1-score across optimization-based RFC models and classical classifiers for 1st dataset.

| Methods | Accuracy (mean ± std) | Precision (mean ± std) | Recall (mean ± std) | F1 Score (mean ± std) |
|---|---|---|---|---|
| RFC with BGWO-FS (5 Wolves) | 0.996 ± 0.008 | 0.996 ± 0.008 | 0.997 ± 0.007 | 0.996 ± 0.008 |
| RFC with BGWO-FS (10 Wolves) | 0.999 ± 0.002 | 0.999 ± 0.002 | 0.999 ± 0.002 | 0.999 ± 0.002 |
| RFC with BGWO-FS (20 Wolves) | 0.999 ± 0.001 | 0.999 ± 0.002 | 0.999 ± 0.001 | 0.999 ± 0.001 |
| RFC with BPSO-FS (5 Swarms) | 0.989 ± 0.004 | 0.989 ± 0.005 | 0.989 ± 0.004 | 0.989 ± 0.004 |
| RFC with BPSO-FS (10 Swarms) | 0.990 ± 0.004 | 0.990 ± 0.004 | 0.990 ± 0.004 | 0.990 ± 0.004 |
| RFC with BPSO-FS (20 Swarms) | 0.991 ± 0.003 | 0.991 ± 0.004 | 0.991 ± 0.003 | 0.991 ± 0.003 |
| RFC with GA-FS (5 Chromosomes) | 0.997 ± 0.002 | 0.997 ± 0.003 | 0.997 ± 0.002 | 0.997 ± 0.002 |
| RFC with GA-FS (10 Chromosomes) | 0.998 ± 0.002 | 0.998 ± 0.003 | 0.999 ± 0.002 | 0.998 ± 0.002 |
| RFC with GA-FS (20 Chromosomes) | 0.997 ± 0.005 | 0.996 ± 0.005 | 0.997 ± 0.005 | 0.997 ± 0.005 |
| Decision Tree | 1.000 ± 0.000 | 1.000 ± 0.000 | 1.000 ± 0.000 | 1.000 ± 0.000 |
| AdaBoost | 0.769 ± 0.000 | 0.769 ± 0.000 | 0.768 ± 0.000 | 0.768 ± 0.000 |
| RFC | 0.824 ± 0.004 | 0.825 ± 0.004 | 0.823 ± 0.004 | 0.823 ± 0.004 |
| Histogram Gradient Boosting | 0.903 ± 0.000 | 0.903 ± 0.000 | 0.902 ± 0.000 | 0.902 ± 0.000 |
| KNN | 0.722 ± 0.000 | 0.722 ± 0.000 | 0.722 ± 0.000 | 0.722 ± 0.000 |
| Gaussian NB | 0.798 ± 0.000 | 0.798 ± 0.000 | 0.798 ± 0.000 | 0.798 ± 0.000 |
| SVM | 0.769 ± 0.011 | 0.770 ± 0.011 | 0.768 ± 0.011 | 0.768 ± 0.011 |
| MLP | 0.937 ± 0.004 | 0.937 ± 0.004 | 0.937 ± 0.004 | 0.937 ± 0.004 |
| DNN | 0.793 ± 0.015 | 0.795 ± 0.016 | 0.792 ± 0.015 | 0.792 ± 0.015 |

**Table 4.** Comparison of testing accuracy, precision, recall, and F1-score across optimization-based RFC models and classical classifiers for 1st dataset.

| Method | Accuracy (mean ± std) | Precision (mean ± std) | Recall (mean ± std) | F1 Score (mean ± std) |
|---|---|---|---|---|
| RFC with BGWO-FS (5 Wolves) | 0.747 ± 0.011 | 0.770 ± 0.012 | 0.724 ± 0.012 | 0.746 ± 0.011 |
| RFC with BGWO-FS (10 Wolves) | 0.748 ± 0.007 | 0.772 ± 0.008 | 0.725 ± 0.008 | 0.748 ± 0.006 |
| RFC with BGWO-FS (20 Wolves) | 0.748 ± 0.008 | 0.772 ± 0.009 | 0.725 ± 0.009 | 0.747 ± 0.008 |
| RFC with BPSO-FS (5 Swarms) | 0.744 ± 0.011 | 0.768 ± 0.012 | 0.718 ± 0.012 | 0.744 ± 0.007 |
| RFC with BPSO-FS (10 Swarms) | 0.745 ± 0.012 | 0.770 ± 0.013 | 0.720 ± 0.012 | 0.745 ± 0.006 |
| RFC with BPSO-FS (20 Swarms) | 0.747 ± 0.011 | 0.772 ± 0.012 | 0.722 ± 0.011 | 0.747 ± 0.006 |
| RFC with GA-FS (5 Chromosomes) | 0.747 ± 0.007 | 0.770 ± 0.008 | 0.721 ± 0.007 | 0.745 ± 0.004 |
| RFC with GA-FS (10 Chromosomes) | 0.746 ± 0.007 | 0.770 ± 0.008 | 0.720 ± 0.008 | 0.746 ± 0.005 |
| RFC with GA-FS (20 Chromosomes) | 0.745 ± 0.007 | 0.769 ± 0.009 | 0.718 ± 0.008 | 0.745 ± 0.005 |
| Decision Tree | 0.643 ± 0.008 | 0.642 ± 0.008 | 0.643 ± 0.008 | 0.642 ± 0.005 |
| AdaBoost | 0.733 ± 0.000 | 0.733 ± 0.000 | 0.733 ± 0.000 | 0.733 ± 0.000 |
| RFC | 0.748 ± 0.005 | 0.748 ± 0.005 | 0.749 ± 0.005 | 0.748 ± 0.005 |
| Histogram Gradient Boosting | 0.745 ± 0.000 | 0.746 ± 0.000 | 0.746 ± 0.000 | 0.745 ± 0.000 |
| KNN | 0.696 ± 0.000 | 0.696 ± 0.000 | 0.697 ± 0.000 | 0.696 ± 0.000 |
| Gaussian NB | 0.721 ± 0.000 | 0.721 ± 0.000 | 0.720 ± 0.000 | 0.720 ± 0.000 |
| SVM | 0.743 ± 0.000 | 0.744 ± 0.000 | 0.744 ± 0.000 | 0.743 ± 0.000 |
| MLP | 0.712 ± 0.011 | 0.713 ± 0.011 | 0.713 ± 0.011 | 0.712 ± 0.011 |
| DNN | 0.747 ± 0.008 | 0.751 ± 0.009 | 0.743 ± 0.008 | 0.749 ± 0.006 |

**Table 5.** T-test comparison of computation time and F1 score performance between BGWO-FS (10 Wolves) baseline and other classifiers. Significance is evaluated at $p < 0.05$.

| Method | *t* (Time) | *p* (Time) | Time Sig | *t* (F1 Score Test) | *p* (F1 Score Test) | F1 Score Sig |
|---|---|---|---|---|---|---|
| RFC with BGWO-FS (5 Wolves) | -0.3691 | 0.7138 | No | -0.6659 | 0.5093 | No |
| RFC with BGWO-FS (20 Wolves) | -0.6491 | 0.5199 | No | -0.0618 | 0.9510 | No |
| RFC with BPSO-FS (5 Swarms) | 9.0479 | 0.0000 | Yes ↑ | -2.7031 | 0.0095 | Yes ↓ |
| RFC with BPSO-FS (10 Swarms) | 12.0580 | 0.0000 | Yes ↑ | -2.3251 | 0.0244 | Yes ↓ |
| RFC with BPSO-FS (20 Swarms) | 11.6021 | 0.0000 | Yes ↑ | -0.9455 | 0.3493 | No |
| RFC with GA-FS (5 Chromosomes) | 9.9132 | 0.0000 | Yes ↑ | -2.3943 | 0.0212 | Yes ↓ |
| RFC with GA-FS (10 Chromosomes) | 13.0019 | 0.0000 | Yes ↑ | -1.8857 | 0.0657 | No |
| RFC with GA-FS (20 Chromosomes) | 12.3541 | 0.0000 | Yes ↑ | -2.5045 | 0.0159 | Yes ↓ |
| Decision Tree | -9.0876 | 0.0000 | Yes ↓ | -61.4316 | 0.0000 | Yes ↓ |
| AdaBoost | -9.0509 | 0.0000 | Yes ↓ | -11.6492 | 0.0000 | Yes ↓ |
| Random Forest | -9.0505 | 0.0000 | Yes ↓ | -2.3948 | 0.0212 | Yes ↓ |
| Histogram Gradient Boosting | -9.0667 | 0.0000 | Yes ↓ | -2.3744 | 0.0259 | Yes ↓ |
| Gaussian NB | -9.0887 | 0.0000 | Yes ↓ | -38.8553 | 0.0000 | Yes ↓ |
| KNN | -9.0887 | 0.0000 | Yes ↓ | -20.3057 | 0.0000 | Yes ↓ |
| SVM | -9.0750 | 0.0000 | Yes ↓ | -4.2293 | 0.0003 | Yes ↓ |
| MLP | -7.5740 | 0.0000 | Yes ↓ | -14.5796 | 0.0000 | Yes ↓ |
| DNN | -7.9524 | 0.0000 | Yes ↓ | -0.8852 | 0.3806 | No |

**Table 5** shows the empirical evidence from the application of a series of pairwise $t$-tests to 1st dataset. The RFC with BGWO-FS with 10 wolves serves as the baseline. The findings demonstrate that neither the RFC with BGWO-FS with 5 wolves nor with 20 wolves exhibited statistically distinct differences in computation time ($p = 0.7138$ and 0.5199) and F1-score outcome ($p = 0.5093$ and 0.9510), indicating the consistent behaviour of the RFC with BGWO-FS with the different number of wolves. Thus, the evidence verifies that the RFC with BGWO-FS is consistent and strong as the swarm population changes moderately, proving its effectiveness for practical use. Conversely, in all configurations of the RFC with BPSO-FS and the RFC with GA-FS, there are noticeable overall differences in the time taken to make calculations (all $p < 0.0001$) and in these configurations' F1-scores < 0; in some cases, like the RFC with BPSO-FS of 5 swarms ($t = -2.7031, p = 0.0095$) and the RFC with GA-FS of 5 chromosomes ($t = -2.3943, p = 0.0212$), the score is increased. Benchmark classifiers also have statistically worse F1 scores, even with very negative t-statistics (e.g., Decision Tree $t = -61.4316$, Gaussian NB $t = -38.8553$, KNN $t = -20.3057$, all $p < 0.0001$) while training, with KNN training the fastest of these. This illustrates that computational efficiency solved an even bigger problem in medical decision-making. All things considered, these $t$-test results demonstrate that, within the 1st dataset, the RFC with BGWO-FS not only offers the best predictive performance, but also the most consistent and reliable performance, far exceeding the other metaheuristics and classical machine learning approaches with statistically significant results.

The 2nd dataset, results shown in **Tables 6** and **7** confirm that the RFC with BGWO-FS is also the most reliable and consistent in performance in training and testing phases. All the RFC with BGWO-FS models (5, 10, and 20 wolves) configurations recorded complete values of training accuracy, precision, recall, and F1-scores ($1.000 \pm 0.000$) as shown in **Table 6**, which means that the training samples were fully classified without errors. This is not surprising due to the RFC with BGWO-FS's rapid convergence toward optimal feature subsets due to the lower dimensionality and a smaller sample size of 2nd dataset. In contrast and the RFC with GA-FS, and the RFC with BPSO-FS, while training also showed positive results, some configurations, particularly the RFC with GA-FS of 20 chromosomes, showed lower stability with accuracy falling to $0.963 \pm 0.068$, which means that it might be prone to premature convergence and population stagnation. In training, the conventional classifiers like Gaussian NB (0.722), KNN (0.798), and SVM (0.773) had a poor training performance, while Decision Tree and Histogram Gradient Boosting Classifiers had a good training accuracy, but overfitting is a major concern.

**Table 6.** Comparison of training accuracy, precision, recall, and F1-score across optimization-based RFC models and classical classifiers for 2nd dataset.

| Methods | Accuracy (mean ± std) | Precision (mean ± std) | Recall (mean ± std) | F1 Score (mean ± std) |
|---|---|---|---|---|
| RFC with BGWO-FS (5 Wolves) | 1.000 ± 0.000 | 1.000 ± 0.000 | 1.000 ± 0.000 | 1.000 ± 0.000 |
| RFC with BGWO-FS (10 Wolves) | 1.000 ± 0.000 | 1.000 ± 0.000 | 1.000 ± 0.000 | 1.000 ± 0.000 |
| RFC with BGWO-FS (20 Wolves) | 1.000 ± 0.000 | 1.000 ± 0.000 | 1.000 ± 0.000 | 1.000 ± 0.000 |
| RFC with BPSO-FS (5 Swarms) | 1.000 ± 0.000 | 1.000 ± 0.000 | 1.000 ± 0.000 | 1.000 ± 0.000 |
| RFC with BPSO-FS (10 Swarms) | 1.000 ± 0.000 | 1.000 ± 0.000 | 1.000 ± 0.000 | 1.000 ± 0.000 |
| RFC with BPSO-FS (20 Swarms) | 1.000 ± 0.000 | 1.000 ± 0.000 | 1.000 ± 0.000 | 1.000 ± 0.000 |
| RFC with GA-FS (5 Chromosomes) | 1.000 ± 0.000 | 1.000 ± 0.000 | 1.000 ± 0.000 | 1.000 ± 0.000 |
| RFC with GA-FS (10 Chromosomes) | 0.999 ± 0.000 | 0.999 ± 0.000 | 0.999 ± 0.000 | 0.999 ± 0.000 |
| RFC with GA-FS (20 Chromosomes) | 0.963 ± 0.068 | 0.962 ± 0.068 | 0.963 ± 0.068 | 0.963 ± 0.068 |
| Decision Tree | 1.000 ± 0.000 | 1.000 ± 0.000 | 1.000 ± 0.000 | 1.000 ± 0.000 |
| AdaBoost | 0.769 ± 0.000 | 0.769 ± 0.000 | 0.768 ± 0.000 | 0.768 ± 0.000 |
| Random Forest | 0.823 ± 0.003 | 0.824 ± 0.003 | 0.823 ± 0.003 | 0.823 ± 0.003 |
| Histogram Gradient Boosting | 1.000 ± 0.000 | 1.000 ± 0.000 | 1.000 ± 0.000 | 1.000 ± 0.000 |
| KNN | 0.798 ± 0.000 | 0.798 ± 0.000 | 0.798 ± 0.000 | 0.798 ± 0.000 |
| Gaussian NB | 0.722 ± 0.000 | 0.722 ± 0.000 | 0.722 ± 0.000 | 0.722 ± 0.000 |
| SVM | 0.773 ± 0.010 | 0.775 ± 0.010 | 0.770 ± 0.010 | 0.770 ± 0.010 |
| MLP | 0.933 ± 0.005 | 0.934 ± 0.005 | 0.933 ± 0.005 | 0.933 ± 0.005 |
| DNN | 0.794 ± 0.014 | 0.795 ± 0.015 | 0.793 ± 0.015 | 0.793 ± 0.015 |

**Table 7.** Comparison of testing accuracy, precision, recall, and F1-score across optimization-based RFC models and classical classifiers for 2nd dataset.

| Methods | Accuracy (mean ± std) | Precision (mean ± std) | Recall (mean ± std) | F1 Score (mean ± std) |
|---|---|---|---|---|
| RFC with BGWO-FS (5 Wolves) | 0.746 ± 0.011 | 0.770 ± 0.012 | 0.724 ± 0.012 | 0.746 ± 0.011 |
| RFC with BGWO-FS (10 Wolves) | **0.748 ± 0.007** | **0.772 ± 0.008** | **0.725 ± 0.008** | **0.748 ± 0.006** |
| RFC with BGWO-FS (20 Wolves) | 0.748 ± 0.008 | 0.772 ± 0.009 | 0.725 ± 0.009 | 0.747 ± 0.008 |
| RFC with BPSO-FS (5 Swarms) | 0.746 ± 0.010 | 0.770 ± 0.009 | 0.724 ± 0.010 | 0.746 ± 0.010 |
| RFC with BPSO-FS (10 Swarms) | 0.746 ± 0.010 | 0.770 ± 0.011 | 0.724 ± 0.010 | 0.746 ± 0.010 |
| RFC with BPSO-FS (20 Swarms) | 0.737 ± 0.013 | 0.760 ± 0.014 | 0.714 ± 0.014 | 0.736 ± 0.013 |
| RFC with GA-FS (5 Chromosomes) | 0.735 ± 0.011 | 0.758 ± 0.011 | 0.746 ± 0.011 | 0.735 ± 0.011 |
| RFC with GA-FS (10 Chromosomes) | 0.738 ± 0.011 | 0.760 ± 0.012 | 0.716 ± 0.011 | 0.737 ± 0.011 |
| RFC with GA-FS (20 Chromosomes) | 0.708 ± 0.139 | 0.728 ± 0.156 | 0.687 ± 0.144 | 0.693 ± 0.139 |
| Decision Tree | 0.644 ± 0.008 | 0.644 ± 0.008 | 0.644 ± 0.008 | 0.644 ± 0.008 |
| AdaBoost | 0.733 ± 0.000 | 0.733 ± 0.000 | 0.733 ± 0.000 | 0.733 ± 0.000 |
| Random Forest | 0.747 ± 0.005 | 0.749 ± 0.005 | 0.748 ± 0.005 | 0.747 ± 0.005 |
| Histogram Gradient Boosting | 0.736 ± 0.000 | 0.714 ± 0.000 | 0.728 ± 0.000 | 0.718 ± 0.000 |
| KNN | 0.696 ± 0.000 | 0.696 ± 0.000 | 0.697 ± 0.000 | 0.696 ± 0.000 |
| Gaussian NB | 0.721 ± 0.000 | 0.721 ± 0.000 | 0.720 ± 0.000 | 0.720 ± 0.000 |
| MLP | 0.715 ± 0.010 | 0.716 ± 0.010 | 0.716 ± 0.010 | 0.715 ± 0.010 |
| SVM | 0.726 ± 0.010 | 0.727 ± 0.010 | 0.725 ± 0.010 | 0.725 ± 0.010 |
| DNN | 0.744 ± 0.011 | 0.747 ± 0.012 | 0.744 ± 0.012 | 0.744 ± 0.011 |

**Table 7** illustrates the performance advantages of the RFC with BGWO-FS had over the other 2nd datasets. All the RFC with BGWO-FS models configurations recorded comparable values in test accuracy (0.746–0.748), precision values (0.770–0.772), and F1-scores (0.746–0.748). The narrow difference in performance shows the consistency of predictions BGWO-FS had over a small and potentially noisy dataset. Most of the values were also stable comparable to the values of BPSO-FS and GA-FS, and in many instances of better, which was the case for the RFC with BPSO-FS with 20 swarms which dropped to 0.736 for the accuracy record and GA-FS with 20 chromosomes dropped even more significantly to 0.693 ± 0.139 F1-score, thus indicating the case of overfitting and unstable values when the searching optimization was too flexible. Being compared to the classical machine learning methods, the RFC with BGWO-FS in also

demonstrated rigorously better performance in comparison to also meeting the other benchmarks, in the 2nd dataset. The RFC with BGWO-FS was better than the 0.747 Random Forest baseline, reasonable, but AdaBoost (0.733), DNN (0.744), and 0.733 was better than other models such as Gaussian NB, KNN, MLP. To conclude, the RFC with BGWO-FS presented the best-balanced reliable performance in terms of generalization in the 2nd dataset and exhibited better accuracy, precision, recall, and F1-score.

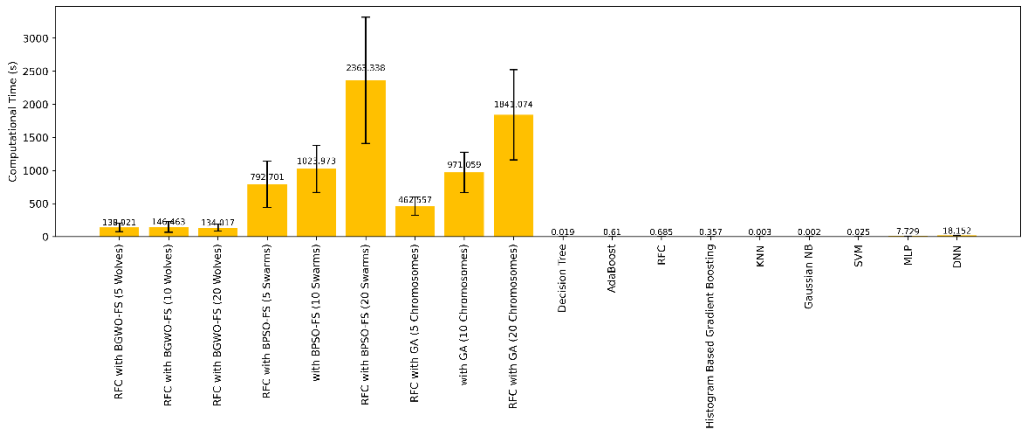
Dual records can be shown in **Table 8** through the paired $t$-tests approach in which the RFC with BGWO-FS of 10 wolves served as the baseline in the 2nd dataset. The performance of the RFC with BGWO-FS of 5 and 20 wolves was not significantly different than the baseline because of the computation time F1-score ($p = 0.0703$ and 0.3779), confirming that the RFC with BGWO-FS was stable for less population values. The configurations of the RFC with BPSO are also the only ones that demonstrate significantly lower F1 performances (e.g., the RFC with BPSO-FS with 5 swarms: $t = -5.0890, p = 6.016 \times 10^{-6}$) and are computationally slower than the rest (all $p < 10^{-11}$). This shows that BPSO-FS is statistically worse than the RFC with BGWO-FS for 2nd dataset. Similarly, GA-FS all configurations take significantly more time to compute (all $p < 10^{-12}$) and the majority exhibit substantially weaker F1's with the exception of GA-FS with 20 chromosomes in which the difference is statistically non-significant ($p = 0.1437$).

**Table 8.** T-test comparison of computation time and F1 score performance between BGWO-FS (10 Wolves) baseline and other classifiers. Significance is evaluated at $p < 0.05$.

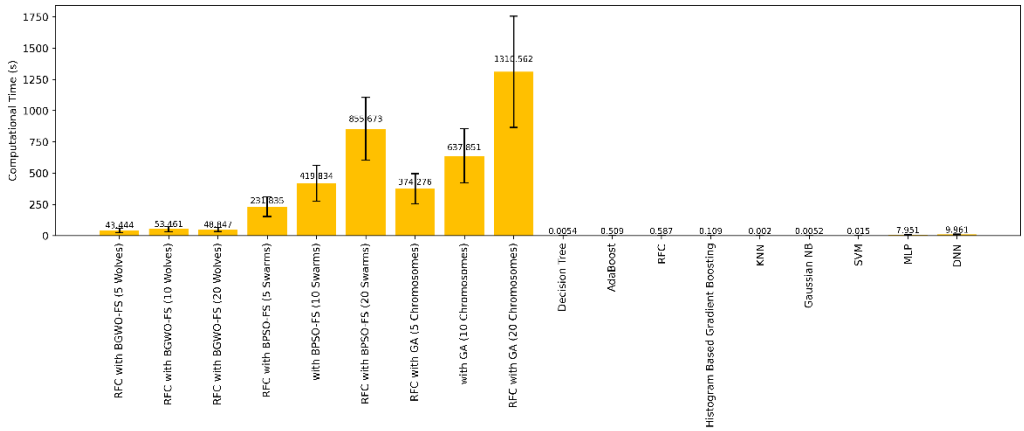
| Method | *t*(Time) | *p*(Time) | Time Sig | *t*(F1 Score Test) | *p*(F1 Score Test) | F1 Score Sig |
|---|---|---|---|---|---|---|
| RFC with BGWO-FS (5 Wolves) | -1.8524 | 0.0703 | = | -1.2247 | 0.2274 | = |
| RFC with BGWO-FS (20 Wolves) | -0.8907 | 0.3779 | = | -0.1067 | 0.9155 | = |
| RFC with BPSO (5 Swarms) | 10.9702 | 1.615e-11 | ↑ slower | -5.0890 | 6.016e-06 | ↓ worse |
| RFC with BPSO (10 Swarms) | 12.5843 | 2.580e-12 | ↑ slower | -5.1327 | 5.135e-06 | ↓ worse |
| RFC with BPSO (20 Swarms) | 16.0057 | 2.018e-14 | ↑ slower | -3.4965 | 0.0011 | ↓ worse |
| RFC with GA (5 Chromosomes) | 12.9980 | 1.001e-12 | ↑ slower | -2.8495 | 0.0067 | ↓ worse |
| RFC with GA (10 Chromosomes) | 16.0044 | 1.993e-14 | ↑ slower | -3.4411 | 0.0012 | ↓ worse |
| RFC with GA (20 Chromosomes) | 14.7412 | 3.629e-14 | ↑ slower | -1.5070 | 0.1437 | = |
| Decision Tree | -12.8730 | 2.881e-12 | ↓ faster | -21.4856 | 8.015e-26 | ↓ much worse |
| AdaBoost | -12.7515 | 3.513e-12 | ↓ faster | -12.3536 | 6.835e-12 | ↓ much worse |
| Random Forest | -12.7678 | 3.419e-12 | ↓ faster | -5.6119 | 1.113e-06 | ↓ worse |
| Histogram Gradient Boosting | -12.8480 | 3.002e-12 | ↓ faster | -15.9470 | 2.842e-14 | ↓ much worse |
| Gaussian NB | -12.8731 | 2.881e-12 | ↓ faster | -13.9987 | 4.838e-13 | ↓ much worse |
| KNN | -12.8738 | 2.878e-12 | ↓ faster | -50.0076 | 8.813e-26 | ↓ much worse |
| SVM | -12.8691 | 2.900e-12 | ↓ faster | -16.2771 | 1.809e-14 | ↓ much worse |
| MLP | -12.3373 | 6.568e-12 | ↓ faster | -21.0248 | 7.487e-26 | ↓ much worse |
| DNN | -10.3738 | 1.558e-10 | ↓ faster | -8.8239 | 4.894e-11 | ↓ worse |

The benchmark classifiers also show clear and consistent statistical inferiority in F1 performance. All the traditional models demonstrate large negative t-statistics coupled with very small p-values, Decision Tree ($t = -21.4856, p = 8.015\,x\,10^{-26}$), Gaussian NB ($t = -13.9987, p = 4.838 \times 10^{-13}$), KNN ($t = -50.0076, p = 8.813 \times 10^{-26}$), SVM ($t = -16.2771, p = 1.809 \times 10^{-14}$), illustrating that significantly weaker predictive capabilities are present. These models also demonstrate statistical superiority in training runtime as indicated $t(Time) < 0$ and $p < 10^{-12}$. These results as a whole also go to demonstrate that predictive performance is worse, confirming a difference in kind underlying the need for a clearly superior method which is an accuracy-driven method rather than a speed-driven model for such sensitive fields. Considering that this is the 2nd dataset and that all t-tests confirmed BGWO-FS is the only method which achieves significant results in terms of stability and efficiency, the results can also be confirmed from a statistical point of view when compared to the heuristics.

The advantage of the RFC with BGWO-FS is supported by Operationalization of Analytics. The RFC with BGWO-FS shows drastically lower execution times than the RFC with BPSO-FS and the RFC with GA-FS, while executing superior predictions as shown in **Figures 4 (a)** and **(b)**. Additionally, the disparate the RFC with BGWO wolf population sizes do not translate to statistical differences involving accuracy, F1-score, or computational time (*t*-tests results shown in **Tables 5** and **8**) showcasing the method's operational stability. Conversely, lower of the stats is shown from the RFC with GA-FS and the RFC with BPSO-FS across configurations. The RFC with BGWO-FS also shows clarity of the advantage as compared to traditional ML methods. The benchmark classifiers (Gaussian NB, KNN, SVM, and MLP) show F1-score undercut and unfit heavily, while showing Decision Tree and Histogram Gradient Boosting extreme overfitting with perfect training accuracy. Balance and reliability stand, making BGWO-FS the best approach among the rest regarding DM prediction across Mixtures datasets.




(a)

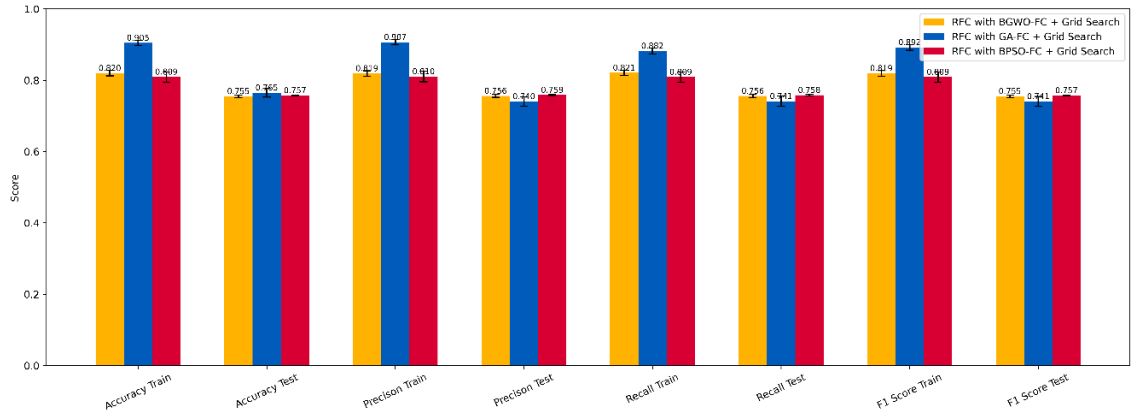



(b)

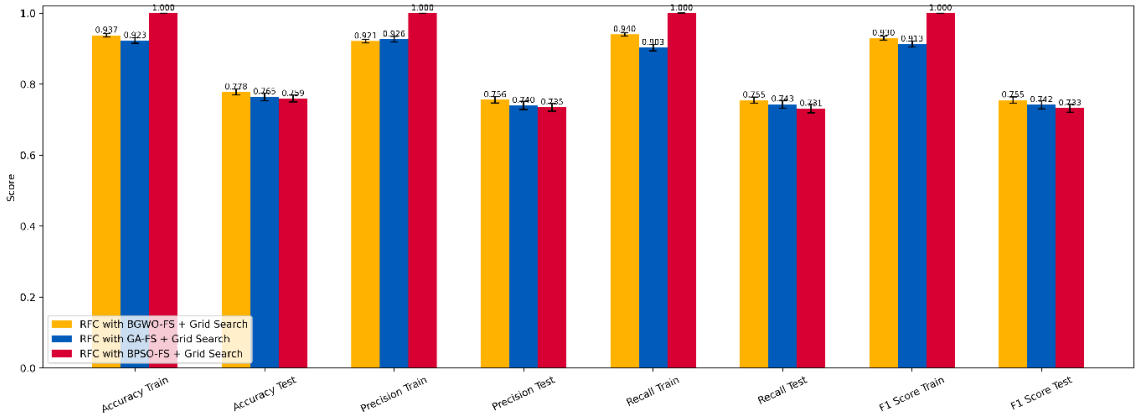
**Figure 4.** Comparison of the computational time required for the DM prediction method using the RFC with BGWO-FS and the benchmark methods to converge. (a) 1st Dataset. (b) 2nd Dataset.

As shown in **Figure 4(a)**, the computational time is compared for the 1st dataset, depicting training costs differences across the methods. The fastest classifiers are those in the KNN, Gaussian NB, and Decision Tree family, which took between 0.002 and 0.019 seconds to respond. Ensemble based baselines, in the

form of AdaBoost, Random Forest, and Histogram Gradient Boosting, also experienced a low compute time of under one second. On the other hand, of the standard models, the MLP and DNN implementations are the slowest, taking 7.7 seconds and 18.2 seconds respectively. However, the metaheuristic-based methods are also much slower, the RFC with BGWO-FS has the best execution time, with a range of 134 to 146 seconds. The RFC with BPSO-FS takes much longer in the range of 792 to 2363 seconds, and RFC with GA-FS is the slowest of the three at a range of 462 to 1841 seconds. In regard to the results for 2nd dataset shown in **Figure 4(b)**, we note that the overall training times are somewhat smaller owing to the smaller size of the training set. We note that the training of the traditional classifiers is instantaneous whereas the training of the MLP and the DNN consume a considerable amount of training time (8–10 s). Among the optimization-based procedures, the RFC with BGWO-FS has the least training time. The RFC with BPSO-FS has relatively high training times (231–855 s), while the RFC with GA-FS is the slowest (374–1310 s).




(a)




(b)

**Figure 5.** The average of performance comparison of standard RFC, the RFC with BGWO-FS, the RFC with BPSO-FS and the RFC with GA-FS optimized using grid search across training and test sets.

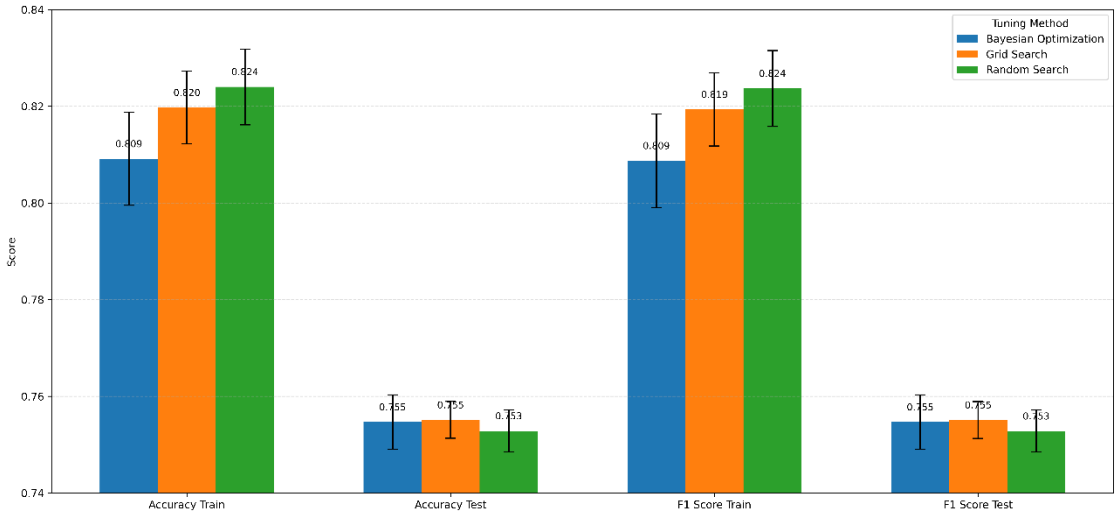
The computational overhead of the the RFC with BGWO-FS is, in practice, significant and, while the RFC with BGWO is accurate and robust, and it is a reasonable fit for offline processing or batch processing. Not in the the RFC with BGWO, of models, sustained accelerators, and over processors. The foregoing the RFC with BGWO-FS is a good fit in terms of trade-offs for efficiency and accuracy and, with reasonable

adjustments, is suitable for deployment in real-life healthcare scenarios. In an effort to address overfitting, a General Grid Search hyperparameter tuning approach was employed to enhance the generalization of the models. In the comparative analysis, RFC was optimized using the four tuning techniques, namely General Grid Search, the RFC with GA-FS, the RFC with BPSO-FS, and the RFC with BGWO-FS. The training and testing results were evaluated using four classification performance metrics, accuracy, precision, recall, and F1-score, as summarized in **Figure 5**.

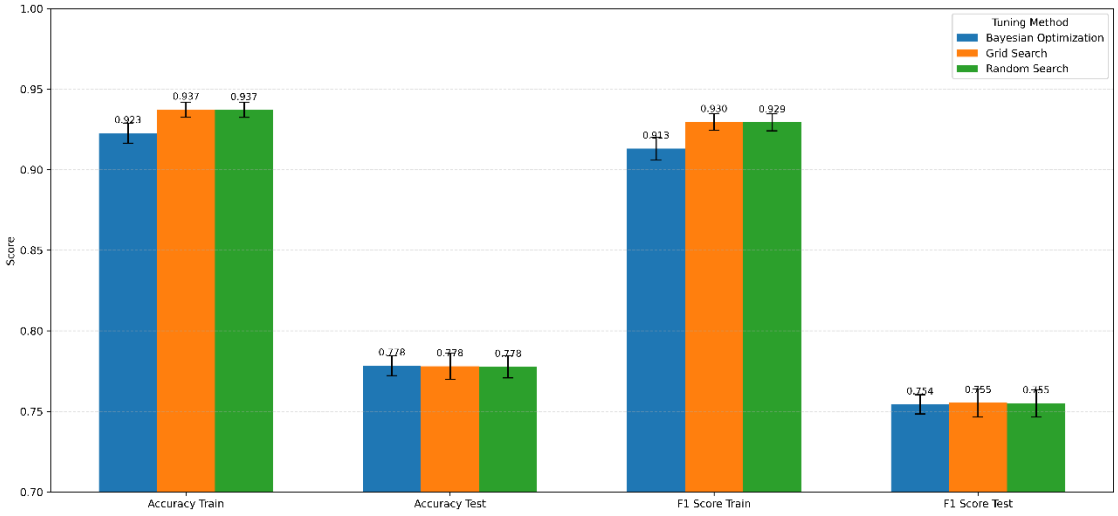
Considering both datasets, the RFC with BGWO-FS consistently achieves the highest and most consistent performance across the benchmark and optimization methods on the test set. This configuration's outperforming can be attributed to RFC's dramatically enhanced capability to generalize as a result of integrating BGWO-FS and thus perform consistently on novel datasets. The performance boost as a result of incorporating Grid Search is attributed to tuning the hyperparameters as well as curtailing the overfitting associated with RFC. The benchmark RFC with Grid Search observed an impressive absolute training score on both datasets, with a comparative dip in the test set score indicating overfitting the model coupled with a lack of robustness. Further, the RFC with GA-FS and BPSO-FS, while test metrics were improved albeit elementary in generalization, consistently achieved the RFC with BGWO-FS outperformance. Both methods exhibited notable overfitting symptoms as evident in the 2nd dataset where they achieved a performance plateau on the training set with a subsequent notable reduction in performance on the test set. Nonetheless, the RFC with GA-FS, the RFC with BPSO-FS, and the baseline RFC showed less overfitting, although the RFC with BGWO-FS did show overfitting, the gap in performance within the training and test set was much smaller. This suggests that the RFC with BGWO-FS is better at recognizing the most relevant and generalizable feature subsets without overfitting the training dataset.

Among all the optimization techniques and the classic classifiers, the RFC with BGWO-FS in both datasets outperform all competitors, and although overall the results in the **Figures 6** to **8** of the model are considerably accurate, do indicate that a gap between train and test accuracy exists, showing that this is not the general case and that moderate overfitting is present. In the 2$^{nd}$ dataset, this is certainly the case, as the model achieves a perfect accuracy of 1.000 in the training data, which is the only class that is overfitted, while the accuracy of the test data only touches between 0.746-0.748, while all the others show a more stable optimization with more generalization than the other two, which are the RFC with GA-FS and the RFC with BPSO-FS.

**Figure 6** illustrates the grouped accuracy and F1-score comparisons across the three hyperparameter tuning strategies—Bayesian Optimization, Grid Search, and Random Search—versus the RFC with BGWO-FS on 1$^{st}$ dataset and 2$^{nd}$ dataset. Across the two datasets, the results indicate that, regardless of the tuning strategy employed, all tuning strategies indicate excellent performance on the training set, with Random Search and its training F1-scores and accuracy being the highest. However, this excellent training performance does not generalize to the test datasets. On 1st dataset, Bayesian Optimization and Grid Search are slightly better and more stable than Random Search on the test set, indicating that they overfit less. This is interpreted to mean that, although Random Search likely covered more ground than the other strategies, there is a chance they had a tighter fit to the training distribution, due to the training set bias. The same pattern is observed in 2$^{nd}$ dataset. Grid Search and Random Search, having achieved virtually perfect scores on the training set, show a marked decrease in the performance gap on the test set, over the other strategies, and all three strategies demonstrate a stable performance there, on the test set. This indicates that BGWO-FS has a different degree of fundamental operational flexibility with respect to the hyperparameter optimizing methods employed. Bayesian Optimization illustrates still more generalization even with the more conservative training performance, while at the same time, Grid Search provides the best compromise between training fit and testing robustness, time and time again.




(a)

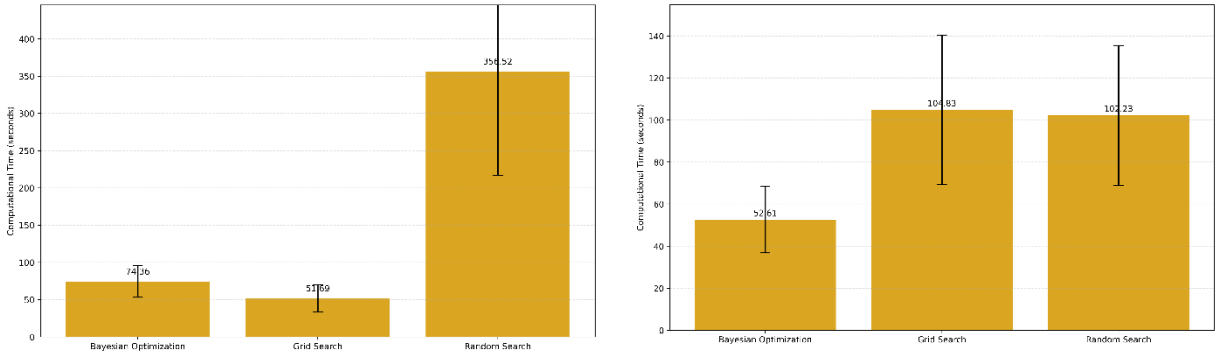



(b)

**Figure 6.** Grouped comparison of accuracy and F1-Score for the RFC with BGWO-FC using three hyperparameter tuning strategies (Bayesian optimization, grid search, and random search). (a) 1st dataset, (b) 2nd dataset.

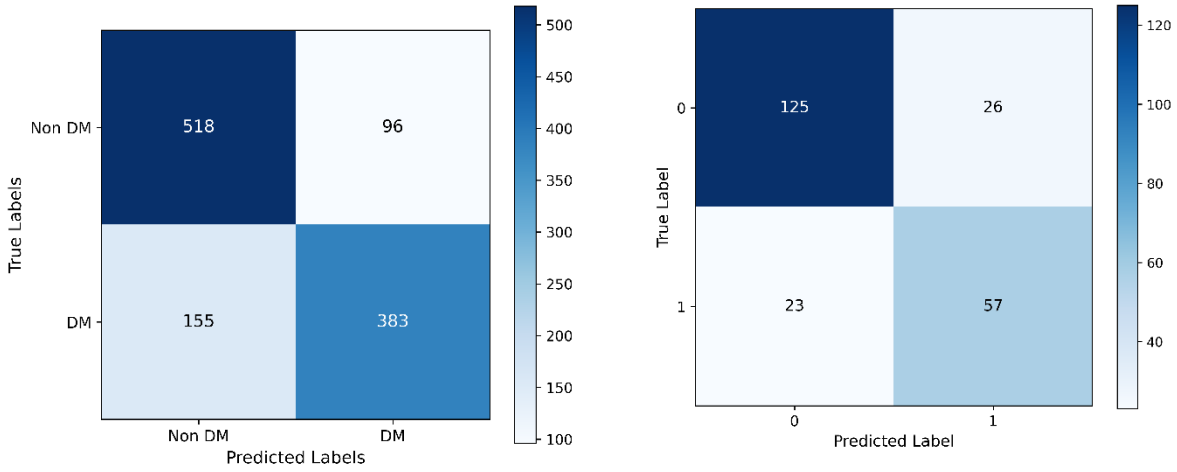
Further evidence for this claim has already been illustrated in **Figure 7**, where we compare the three tuning strategies for their computational time. As such, for instance, in terms of the time taken, Bayesian Optimization has the lowest mean computation time, in this case being 52.61 seconds of time for the run, making it the more computationally efficient alternative. On the other hand, the other two strategies, Grid Search, and Random Search, took almost twice of that of Bayesian Optimization (104.83 and 102.23 seconds respectively), and for that reason ended up being less efficient. Even with this, Grid Search has more stable and consistent performance, and for this reason also has the higher computational cost. On the other hand, Bayesian Optimization had higher reliability when it came the performance and generalization. In conclusion, we can say that the RFC with BGWO-FS would lead to consistent performance, thus obtaining reliability with lower cost of computational power. Considering all of this, we can say that Grid

Search shows the highest reliability for the datasets while at the same time being the most generalizable approach while obtaining stable performance within the datasets. In Bayesian Optimization, it had competitive performance but lower computational cost making it efficient when compared to Random Search. Even with the same computational cost, Random Search provided a less tuneable performance more differentiated for training and testing with the higher datasets, primarily the one with the higher number of datasets. This just shows the RFC with BGWO-FS system while also showing the clear difference it makes with the hyper parameter selection for the tuning strategy recommended for the medical predictive modelling tasks.



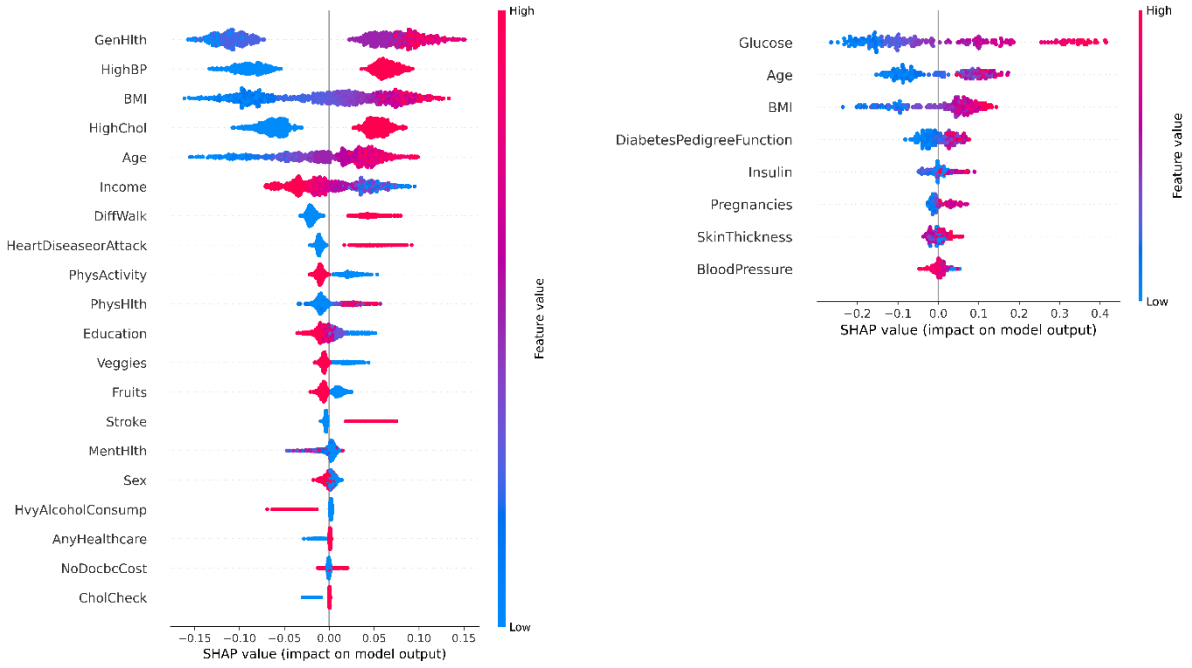

(a) (b)

**Figure 7.** Comparison of computational time among Bayesian Optimization, Grid Search, and Random Search. (a) 1st dataset, (b) 2nd dataset.




(a) (b)

**Figure 8.** Confusion matrix for the DM prediction method by the RFC with BGWO-FS and Grid Search on test data. (a) 1st dataset, (b) 2nd dataset.




(a) (b)

**Figure 9.** SHAP summary plot for DM prediction by the RFC with BGWO-FS and grid search. (a) 1st dataset. (b) 2nd dataset.

The confusion matrices in **Figure 8** partly suggest that the model demonstrates adequate discriminatory ability in differentiating diabetic and non-diabetic persons. With a more granular examination, however, the model demonstrates apparent challenges in accurately ascertaining the positive diabetic cases. This indicates the model does not do a good job in sufficient positive identification or high recall. This lack of recall is in the resultant added false negatives, where the true positive case of diabetes is missed, and the model is classified as non-diabetic. This effect becomes even more glaring in the clinical setting, as unobserved positive cases cannot be captured, leading to late diagnosis, in the event missing timely diagnosis and undesirable health consequences. Thus, for positive diabetic case identification, it is of utmost importance to have recall performance as a priority in enhancing clinical utility of the model for timely screening and risk triaging. **Figure 8(a)** shows the confusion matrix associated with the first test dataset for the model trained with RFC, BGWO-FS, and Grid Search.

The predictive model identified 518 diabetic and 383 diabetic cases (true positives) and 96 diabetic cases without any medical indication, which were diagnosed as falsely diabetic and pertaining to non-diabetic (false positives); it also had 155 diabetic cases (false negatives) and missed 96 non-diabetic cases. The results also indicated that 78.2% accuracy, 79.9% precision, 71.2% recall, and 75.3% F1-score, respectively. A precision that high meant that the majority of the cases that were diagnosed as diabetic were indeed so, while the recall demonstrated that the model had a great number of false negatives. **Figure 8(b)** shows a second test dataset confusion matrix, wherein the model had a true negative classification of 125, true positive of 57, while also possessing 26 false positive and 23 false negative, amounts, respectively. This also indicated 78.8% accuracy, 68.7% precision, 71.3% recall, and 70.0% F1 score. The recall also coincides with the two datasets, while why recall differs is due to differing amounts of true positive and false positive cases dependent on the dataset. Overall, **Figures 8(a)** and **8(b)** relay that the RFC with

BGWO-FS and Grid Search exhibits consistent and balanced results regardless of the test dataset applied. The model is capable of high levels of discrimination, however, improvements on recall and thus false negatives, are the metrics that will prove most beneficial. Increasing sensitivity would improve the model fit for clinical screening and clinical decision support systems.

Moving to the DM classification, the predictions of the model are explained using SHAP analysis, and in particular, the contribution of the features in the two datasets is demonstrated in the SHAP summary plots of **Figure 9**. Specifically, in **Figure 9**, the higher the GenHlth, HighBP, and BMI, and HighChol, the more SHAP values are positive and the more diabetes is predicted. These features had a relatively large spread among the SHAP contributions, signalling that these features had a more significant impact on the model. These results also reflect the evidence stated in previous studies, which identify poor health conditions, hypertension, obesity, and high cholesterol levels as the main risk factors for diabetes. Other features like Age, Income, DiffWalk, and HeartDiseaseorAttack also added value and these were demonstrated with narrow SHAP ranges, signalling that these features added secondary importance to the likelihood of diabetes in the studied population.

In 2$^{nd}$ dataset shown in **Figure 9(b)**, there is a unique glucose pattern output. Higher glucose values are good indicators of positive SHAP glucose values and are therefore diagnostic in reviewing diabetes clinical criteria. Age / BMI and Diabetes Pedigree Function are core contributors, and are a reminder of the significance in the obesity, metabolic decline and age-related decline. Other features such as Insulin, Pregnancies, Skin Thickness, and Blood Pressure are shown to have small SHAP contributions explaining their slight, if at all, supportive involvement in the pathways associated with diabetes risk. The results, conversely, display the significant influence of other physiological risks.

The first reason that can be attributed to it is the process of incorporating the RFC with BGWO to create a single feature that is more than usually irrefutable; hence the models can focus more on clinically significant differential predictors. The RFC with BGWO is more efficient in focusing on clinically differentiated metrics, thus enhancing stability. This ability helps RFC model patient data heterogeneity and discover more granular risk patterns pertaining to diabetes mellitus. However, there are some shortcomings of the RFC with BGWO-FS. The model shows some overfitting characterised by large gaps in the training versus testing accuracy. This indicates RFC is still learning noise in the dataset instead of entirely learning the more generalizable patterns. Also, model recall is lower than one would prefer, and consequently the model is less able to identify some of the positive DM cases. This is concerning in clinical settings, where lower sensitivity is more dangerous, as it results in cases which remain undiagnosed and leads to delays in treatment. Nonetheless, the RFC with BGWO-FS should have more versatile applications. The model's ability to work with large feature sets makes it a candidate for use in fields such as bioinformatics, financial risk modelling and medical diagnostics. This points to a strong platform for the development of future mixed models. The merge of the RFC with BGWO-FS and other new methodologies such as boosting and deep neural networks or multi-objective optimization is likely to increase forecasting accuracy and streamline computing systems.

Future works ought to focus on incorporating this system into actual clinical settings with a Clinical Decision Support System (CDSS). This system would use Electronic Health Records (EHRs) or user provided data for predicting DM risk automatically. The RFC with BGWO-FS would pinpoint the essential features, allowing the system to run efficiently without sacrificing accuracy. The RFC would output risk predictions and alongside it, offer various confidence levels which could be displayed on a user-friendly clinical dashboard. This would assist healthcare practitioners in early decision making and provide tailored recommendations depending on the risk levels of the patient.

## 5. Conclusions

The research further improved the prediction of DM through the combination of the RFC and BGWO-Feature Selection (RFC with BGWO-FS). Two independent datasets were used to assess the method's performance and it proved to be better than other models utilizing machine learning and other feature selection techniques. In addition to reducing the number of features needed and increasing the accuracy and F1-score of the model, the stability of the model was improved. Tuning hyperparameters improved the overall performance of the model. The features selected through the SHAP analysis were found to be predictors of DM risk and therefore the model's prediction was validated. In this study, the overall performance of the method was found to be high but the model was not able to detect all of the DM positive cases. Because of this, the researchers hope to improve the model's sensitivity and incorporate it with other deep learning frameworks to boost the accuracy of the model. The results of the study demonstrated the efficient, robust and interpretable method of prediction DM with the RFC with BGWO-FS. The method holds promise to being used in early screening of DM and in clinical decision support systems.

### Conflicts of Interest

There are no conflicts of interest related to the publishing of this work, according to the authors.

### Acknowledgments

The authors appreciate the Ministry of Higher Education, Science, and Technology of the Republic of Indonesia for the support of this study. This research was funded (with contract number 00309.103/UN10.A0501/B/PT.01.03.2/2024) by the Directorate General of Higher Education, Research and Technology (DIKTI) as part of the 2024 Fundamental Research Grant.

### AI Disclosure

During the preparation of this work the author(s) used generative AI in order to improve the language of the article. After using this tool/service, the author(s) reviewed and edited the content as needed and take(s) full responsibility for the content of the publication.

## References

Alzboon, M.S., Al-Batah, M.S., Alqaraleh, M., Abuashour, A., & Bader, A.F.H. (2023). Early diagnosis of diabetes: a comparison of machine learning methods. *International Journal of Online and Biomedical Engineering, 19*(15), 144-165. https://doi.org/10.3991/ijoe.v19i15.42417.

Ang, K.H., Ang, K.M., Juhari, M.R.B.M., Wong, C.H., Sharma, A., Ang, C.K., Tiang, S.S., & Lim, W.H. (2023). Classification of wafer defects with optimized deep learning model. *The 2023 International Conference on Artificial Life and Robotics*, *28*, 609-614. https://doi.org/10.5954/ICAROB.2023.OS25-4.

Ang, K.M., Natarajan, E., Isa, N.A.M., Sharma, A., Rahman, H., Then, R.Y.S., Alrifaey, M., Tiang, S.S., & Lim, W.H. (2022). Modified teaching-learning-based optimization and applications in multi-response machining processes. *Computers & Industrial Engineering, 174*, 108719. https://doi.org/10.1016/j.cie.2022.108719.

Ansyari, M.R., Mazdadi, M.I., Indriani, F., Kartini, D., & Saragih, T.H. (2023). Implementation of random forest and extreme gradient boosting in the classification of heart disease using particle swarm optimization feature selection. *Journal of Electronics, Electromedical Engineering, and Medical Informatics, 5*(4), 250-260. https://doi.org/10.35882/jeeemi.v5i4.322.

Behera, S.K., & Mohanty, N.K. (2019). Congestion management using thyristor-controlled series compensator employing improved grey wolf optimization technique. *International Journal of Electrical Engineering & Education, 58*(2), 179-199. https://doi.org/10.1177/0020720918822730.

Breiman, L. (2001). Random forests. *Machine Learning, 45*(1), 5-32. https://doi.org/10.1023/A:1010933404324.

Brezočnik, L., Fister, I., & Podgorelec, V. (2018). Swarm intelligence algorithms for feature selection: a review. *Applied Sciences, 8*(9), 1521. https://doi.org/10.3390/app8091521.

Burgess, J., de Bezenac, C., Keller, S.S., Frank, B., Petropoulos, I.N., Garcia-Finana, M., Jackson, T.L., Kirthi, V., Cuthbertson, D.J., Selvarajah, D., Tesfaye, S., & Alam, U. (2024). Brain alterations in regions associated with end-organ diabetic microvascular disease in diabetes mellitus: a UK biobank study. *Diabetes/Metabolism Research and Reviews, 40*(2), e3772. https://doi.org/10.1002/dmrr.3772.

Kangra, K., & Singh, J. (2024). A genetic algorithm-based feature selection approach for diabetes prediction. *IAES International Journal of Artificial Intelligence, 13*(2), 1489-1498. https://doi.org/10.11591/ijai.v13.i2.pp1489-1498.

Kong, L., & Ma, X. (2018). Comparison study on the nonlinear parameter optimization of nonlinear grey Bernoulli model (NGBM(1,1)) between intelligent optimizers. *Grey Systems: Theory and Application, 8*(2), 210-226. https://doi.org/10.1108/gs-01-2018-0005.

Kordon, A.K. (2010). *Swarm intelligence: the benefits of swarms*. Springer, Berlin, Heidelberg. ISBN: 978-3-540-69913-2. https://doi.org/10.1007/978-3-540-69913-2_6.

Liu, J., Wei, X., & Huang, H. (2021). An improved grey wolf optimization algorithm and its application in path planning. *IEEE Access, 9*, 121944-121956. https://doi.org/10.1109/access.2021.3108973.

Louk, M.H.L., & Tama, B.A. (2022). PSO-driven feature selection and hybrid ensemble for network anomaly detection. *Big Data and Cognitive Computing, 6*(4), 137. https://doi.org/10.3390/bdcc6040137.

Mahmood, I., & Abdullah, H.S. (2024). Analyzing the behavior of different classification algorithms in diabetes prediction. *IAES International Journal of Artificial Intelligence, 13*(1), 201-206. https://doi.org/10.11591/ijai.v13.i1.pp201-206.

Marlina, T.T., Haryani, & Widyawati. (2024). The validity and reliability of the Indonesian version of the diabetes mellitus self-efficacy scale **(**DMSES-I). *Journal of Research in Nursing, 29*(8), 666-678. https://doi.org/10.1177/17449871241276816.

Masud, S.S.B., Mahajan, K., Kondyli, A., Deliali, K., & Yannis, G. (2024). Leveraging machine learning algorithms to predict and analyze single-vehicle and multi-vehicle crash occurrences on motorways. *Transportation Research Record, 2678*(12), 1329-1345. https://doi.org/10.1177/03611981241250348.

Minnoora, M., & Bathsb, V. (2023). Diagnosis of breast cancer using random forests. *Procedia Computer Science, 218*, 429-437. https://doi.org/10.1016/j.procs.2023.01.025.

Mirjalili, S., Mirjalili, S.M., & Lewis, A. (2014). Grey wolf optimizer. *Advances in Engineering Software, 69*, 46-61. https://doi.org/10.1016/j.advengsoft.2013.12.007.

Mohan, S., Thirumalai, C., & Srivastava, G. (2019). Effective heart disease prediction using hybrid machine learning techniques. *IEEE Access, 7*, 81542-81554. https://doi.org/10.1109/ACCESS.2019.2923707.

Nguyen, L.P., Tung, D.D., Nguyen, D.T., Le, H.N., Toàn, T.Q., Binh, T.V., & Pham, D.T.N. (2023). The utilization of machine learning algorithms for assisting physicians in the diagnosis of diabetes. *Diagnostics, 13*(12), 2087. https://doi.org/10.3390/diagnostics13122087.

Nimma, K.S., Al-Falahi, M.D.A., Nguyen, H.D., Jayasinghe, S.D.G., Mahmoud, T.S., & Negnevitsky, M. (2018). Grey wolf optimization-based optimum energy-management and battery-sizing method for grid-connected microgrids. *Energies, 11*(4), 847. https://doi.org/10.3390/en11040847.

Obermeyer, Z., & Emanuel, E.J. (2016). Predicting the future: big data, machine learning, and clinical medicine. *New England Journal of Medicine, 375*(13), 1216-1219. https://doi.org/10.1056/NEJMp1606181.

Olchanski, N., Weidner, S.B., Cohen, J.T., & Kent, D.M. (2024). Value estimation of the diabetes prevention program. *Journal of Clinical and Translational Science, 8*(S1), 116-116. https://doi.org/10.1017/cts.2024.341.

Pan, L., Cheng, W.L., Lim, W.H., Sharma, A., Jately, V., Tiang, S.S., Alherby, A.H., & El-Kenawy, E.S.M. (2025). A robust wrapper-based feature selection technique based on modified teaching-learning-based optimization with hierarchical learning. *Engineering Science and Technology, an International Journal, 61*, 101935. https://doi.org/10.1016/j.jestch.2024.101935.

Papatheodorou, K., Banach, M., Edmonds, M., Papanas, N., & Papazoglou, D. (2015). Complications of diabetes. *Journal of Diabetes Research*, 189525. https://doi.org/10.1155/2018/3086167.

Prakoso, D.A., Istiono, W., Mahendradhata, Y., & Arini, M. (2023). Screening implementation of tuberculosis–diabetes mellitus. *BMC Public Health, 23*(1), 1908. https://doi.org/10.1186/s12889-023-16840-z.

Rodriguez-Galiano, V.F., Ghimire, B., Rogan, J., Chica-Olmo, M., & Rigol-Sanchez, J.P. (2012). An assessment of the effectiveness of a random forest classifier for land-cover classification. *ISPRS Journal of Photogrammetry and Remote Sensing, 67*(1), 93-104. https://doi.org/10.1016/j.isprsjprs.2011.11.002.

Sam'an, M., Farikhin, & Munsarif, M. (2025). An improved decision tree model through hyperparameter optimization using a modified gray wolf optimization for diabetes classification. *Computer Methods in Biomechanics and Biomedical Engineering*, 1-17. https://doi.org/10.1080/10255842.2025.2460178.

Shatnawi, M., Zaki, N., & Yoo, P.D. (2014). Protein inter-domain linker prediction using Random Forest and amino acid physiochemical properties. *BMC Bioinformatics, 15*(16), S8.

Sheta, A., Elashmawi, W.H., Al-Qerem, A., & Othman, E.S. (2024). Utilizing various machine learning techniques for diabetes mellitus feature selection and classification. *International Journal of Advanced Computer Science and Applications, 15*(3), 1372-1384. https://dx.doi.org/10.14569/IJACSA.2024.01503134.

Sirmayanti, Prastyo, P.H., & Mahyati. (2025). Enhancing diabetes prediction performance using feature selection based on grey wolf optimizer with autophagy mechanism. *Computer Methods and Programs in Biomedicine Update, 8*, 100207. https://doi.org/10.1016/j.cmpbup.2025.100207.

Tama, B., & Rhee, K. (2018). An integration of PSO-based feature selection and random forest for anomaly detection in IoT network. *MATEC Web of Conferences, 159*, 01053. https://doi.org/10.1051/matecconf/201815901053.

Xie, Y., Li, X., Ngai, E.W.T., & Ying, W. (2009). Customer churn prediction using improved balanced random forests. *Expert Systems with Applications, 36*(3), 5445-5449. https://doi.org/10.1016/j.eswa.2008.06.121.

Yego, N.K., Kasozi, J., & Nkurunziza, J. (2021). A comparative analysis of machine learning models for the prediction of insurance uptake in Kenya. *Data, 6*(11), 116. https://doi.org/10.3390/data6110116.

Zhang, S., Luo, Q., & Zhou, Y. (2017). Hybrid grey wolf optimizer using elite opposition-based learning strategy and simplex method. *International Journal of Computational Intelligence and Applications, 16*(2), 1750012. https://doi.org/10.1142/S1469026817500122.

Zhao, G., Wang, H., Jia, D., & Wang, Q. (2019). Feature selection of grey wolf optimizer based on quantum computing and uncertain symmetry rough set. *Symmetry*, *11*(12), 1-19. https://doi.org/10.3390/sym11121470.

Zou, Q., Qu, K., Luo, Y., Yin, D., Ju, Y., & Tang, H. (2018). Predicting diabetes mellitus with machine learning techniques. *Frontiers in Genetics, 9*(515), 1-10. https://doi.org/10.3389/fgene.2018.00515.