

Hybrid Binary Grey Wolf Optimizer and Binary Sine Cosine Algorithms for Feature Selection Parkinson Disease

ANDI NUGROHO¹, HARCO LESLIE HENDRIC SPITS WARNARS¹, SANI MUHAMAD ISA²,
 WIDODO BUDIHARTO²

¹Department of Computer Science, Bina Nusantara University, Jakarta, Indonesia

²Department of Business, School of Computer Science, Bina Nusantara University, Jakarta, Indonesia

Corresponding author: Andi Nugroho (e-mail: andi.nugroho002@binus.ac.id).

ABSTRACT The algorithm used in the feature selection process is Hybrid Binary Grey Wolf Optimizer-Binary Sine Cosine Algorithm (HBGWO-BSCA). The dataset used for the HBGWO-BSCA is Parkinson's. This dataset is used because Parkinson's disease is one of the most frequently discussed diseases throughout the world. The aim of this paper is to find out what features are used in the process of predicting Parkinson's disease. The HBGWO-BSC feature selection algorithm was proven to be able to increase the accuracy value of the KNN classification algorithm by 92%, while the HGWO-SCA only obtains an accuracy of 88%. The value of the HBGWO-BSCA is higher than that of HGWO-SCA because the relationship between features selected with the HBGWO BSCA is more accurate than that of the Hybrid Grey Wolf Optimizer-Sine Cosine Algorithm (HGWO-SCA). This proves that the HBGWO-BSC feature selection algorithm obtains the highest accuracy, precision, recall and F1-Score values compared to the HGWO-SCA. The HBGWO-BSCA in feature selection uses parameters N and alpha with parameter value ranges $N=9-11$ and $\alpha=2-8$. HBGWO-BSCA is an algorithm used for the feature selection process in datasets that already have labels or classes.

KEYWORDS Parkinson; HBGWO-BSCA; HGWO-SCA; Feature Selection.

I. INTRODUCTION

Parkinson's disease (PD) is one of the most frequently discussed diseases recently throughout the world. Parkinson's is a progressive neurodegenerative disease associated with impaired motor skills [1–3]. This disease has main symptoms that can be recognized from its characteristics, such as shivering when resting, then slowing body movements, attacks of muscle contractions, abnormalities when the sufferer stands and reflexes that cannot maintain balance [4]. Detecting PD early is very important because the price of the drug is still expensive and not easy to get without a prescription from a doctor. PD, which is a chronic neurodegenerative disease and a progressive neurodegenerative disease is the second most prevalent neurodegenerative disorder after Alzheimer's, and it is associated with damage to nigrostriatal dopaminergic neurons [2, 3]. Early detection of Parkinson's disease allows patients to receive timely treatment, potentially reducing the severity of its symptoms [7–9]. In this research, we will use the PD dataset obtained from the URL address: <https://archive.ics.uci.edu/dataset/174/parkinson> based on research of several scientists including determining appropriate

features to be used in classification algorithms. The feature selection algorithm is used to find which features in the PD dataset are most related to getting the best accuracy, precision, recall and F1-Score values. The feature selection algorithm used in PD is a combination of two feature selection algorithms, namely the Binary Grey Wolf Optimizer (GWO) algorithm and the Binary Sine Cosine Algorithm (BSCA). This algorithm is called the Hybrid Binary Grey Wolf Optimizer-Binary Sine Cosine Algorithm (HBGWO-BSCA). The task of the HBGWO-BSCA is to find the features that are most related to each other so that these features can produce the best accuracy values when used in the classification algorithm. PD is a dataset used to search for features that will later be used in classification algorithms. HBGWO-BSCA will carry out feature selection using the wrapper feature selection method. The wrapper method is a method that is considered the best filter method because this method determines a subset of features depending on the classifier. In addition, the wrapper computing method has a higher level of sensitivity to the classifier, because this method focuses on the classifier that we provide to achieve optimal transfer function values [10]. The

wrapper method is a method that uses directly the classification performance of a data classifier to evaluate the capabilities of a classification algorithm. This method was chosen to achieve better accuracy values in the classification algorithm [11]. The wrapper method is usually used in classification algorithms where the goal is to evaluate a dataset repeatedly from a subset of features selected in the dataset so that features that are interdependent will be selected and those that will not be removed, so that the selected features will be processed in the classification algorithm [12]. HBGWO-BSCA is a hybrid feature selection algorithm that uses the wrapper method. This paper consists of 5 important research sections. Section 2 discusses the research related papers. Section 3 highlights the methods used. Section 4 describes the data processing process and displays the results of the experiments conducted. Section 5 displays and summarizes the results obtained.

II. RELATED WORK

As technology develops, machine learning is now widely used to see, predict and make decisions using machine learning as a reference. That is why the development of machine learning is currently progressing rapidly.

A. FEATURE SELECTION

Feature selection (FS) is the most important procedure in machine learning because FS can reduce the complexity of the model used and can simplify the interpretation of the features to be used [13]. The primary objective of FS is to enhance data pre-processing by reducing the number of features in the dataset, thereby simplifying the data to lower the classification error rate [14]. FS can also be used to represent features in a dataset in a certain problem domain with a subset of features that have been selected by an optimization algorithm to obtain optimal values, so that the resulting accuracy value is higher than without using FS [15]. FS comprises three approaches: the filter method, the wrapper method, and the embedded method [16].

B. FILTER METHOD

The filter method is an FS method that uses feature relevance evaluation through univariate statistics such as correlation coefficient, Fisher score, or mutual information. The feature search filter method performs statistical analysis on a set of features by ranking each feature from a set of data [17].

C. WRAPPER METHOD

The wrapper method is a method that is considered the best filter method because this method determines a subset of features depending on the classifier. In addition, the wrapper computing method has a higher level of sensitivity to the classifier, because this method focuses on the classifier that we provide to achieve the optimal transfer function value [10]. The wrapper method in feature selection uses several approaches such as using several metaheuristic algorithms such as Particle Swarm Optimization (PSO), Dragonfly Algorithm (DA), Grey Wolf Optimizer (GWO), African Buffalo Optimization (ABO), Imperialist Competitive Algorithm (ICA) [18].

D. GREY WOLF OPTIMIZER (GWO)

It integrates a metaheuristic algorithm based on the hierarchical structure and social dynamics exhibited by Grey wolves while hunting their prey [19]. The GWO algorithm is designed based

on the hunting behavior of Grey wolves, where a group of wolves establishes a hierarchy for task assignment during their search for prey [20]. Grey wolves divide four levels of leadership in their hunting group, including: [21].

- Alpha level (α): The Grey wolf seeks the solution with the best objective value in finding prey.
- Beta Level (β): The Grey wolf at this second level is looking for the second best objective value solution in finding prey.
- Delta Level (δ): The Grey wolf at this third level is looking for the third best objective value solution in finding prey.
- Omega Level (ω): This Grey wolf in the fourth level looks for all other best solutions, therefore, three wolves in levels 1,2,3 are the right choice α , β , and δ is responsible for guiding the ω level wolves in determining the prey hunting mechanism.

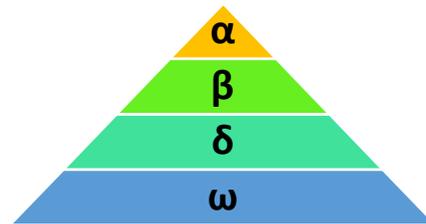


Figure 1. Hierarchical of Grey Wolf Algorithm (GWO) [22].

The siege behavior of Grey wolves in hunting can be expressed mathematically as follows [23]:

$$\vec{D} = \left| \vec{C} \cdot \vec{X}_p(t) - \vec{X}(t) \right|, \quad (1)$$

$$\vec{X}(t+1) = \vec{X}_p(t) - \vec{A} \cdot \vec{D}. \quad (2)$$

For variable t is the number of iterations, then variable \vec{X} is the position of the Grey wolf, vector \vec{X}_p is the vector position of the prey and the variables \vec{A} , \vec{C} , and \vec{D} are the vector coefficient values from the following formula:

$$\vec{A} = 2\vec{a} \cdot \vec{r1} - \vec{a}, \quad (3)$$

$$\vec{C} = 2\vec{r2}, \quad (4)$$

$$\vec{a} = 2 - t \cdot \frac{2}{MaxIter}. \quad (5)$$

Variables $\vec{r1}$ and $\vec{r2}$ have random values from 0, 1, and for variable \vec{a} is the value of subtracting the iteration value from 2 – the value of the iteration variable. Next, based on the formula above, the GWO algorithm has three (3) levels and starts from level one to level three (α , β , and δ) and can be simulated as in the following formula [24]:

$$\begin{cases} \vec{D}_\alpha = \left| \vec{C}_1 \cdot \vec{X}_\alpha - \vec{X} \right| \\ \vec{D}_\beta = \left| \vec{C}_1 \cdot \vec{X}_\beta - \vec{X} \right|, \\ \vec{D}_\delta = \left| \vec{C}_1 \cdot \vec{X}_\delta - \vec{X} \right| \end{cases} \quad (6)$$

$$\begin{cases} \vec{X}_1 = \left| \vec{X}_\alpha - \vec{A}_1 \cdot \vec{D}_\alpha \right| \\ \vec{X}_2 = \left| \vec{X}_\beta - \vec{A}_2 \cdot \vec{D}_\beta \right|, \\ \vec{X}_3 = \left| \vec{X}_\delta - \vec{A}_3 \cdot \vec{D}_\delta \right| \end{cases} \quad (7)$$

$$\vec{X}(t+1) = \frac{\vec{X}_1 + \vec{X}_2 + \vec{X}_3}{3}. \quad (8)$$

The variables $\vec{X}_\alpha, \vec{X}_\beta, \vec{X}_\delta$ indicate the position of the Grey wolf pack from the levels of agent α , agent β and agent δ . Next, the variable \vec{X} shows the set of existing solutions. Then the variables $\vec{X}_1, \vec{X}_2, \vec{X}_3$ show the direction and steps of agent ω towards agent α , agent β and agent δ . The following is the pseudocode of the GWO algorithm based on the formula above.

GWO Pseudocode [25]

```

1 Initialize the Grey wolf population  $X_i (i = 1, 2, \dots, n)$ 
2 Initialize Parameter  $A, a$  dan  $C$ 
3 Calculate the fitness of each search agent
4  $\vec{X}_\alpha \sim$  the best search agent
5  $\vec{X}_\beta \sim$  the second best search agent
6  $\vec{X}_\delta \sim$  the third best search agent
7 While  $t < \text{Max number of iterations}$ 
8   For each search agent
9     Update the position of the current search agent by equation 6, 7, 8
10  End For
11 update parameter GWO ( $A, a$  and  $C$ )
12 Calculate the fitness of all search agents
13 Update  $\vec{X}_\alpha, \vec{X}_\beta, \vec{X}_\delta$ 
14  $t = t + 1$ 
15 End While
16 Return  $\vec{X}_\alpha$ 

```

The pseudocode above is an algorithm from GWO, but over time the GWO algorithm has changed with updates to the GWO algorithm called the Binary Grey Wolf Optimizer (BGWO) algorithm. BGWO Algorithm Optimization Algorithm is used to handle mathematical calculation optimization problems and find the best features [21]. The optimization algorithm requires the identification of two main components, namely the evaluation function (suitability) and the solution representation. The primary aim of feature search is to identify a minimal subset of features, reducing memory and processor usage while maximizing accuracy in the classification algorithm. Therefore, feature selection is known as a multi-objective optimization problem [26].

The BGWO algorithm is included in the feature selection optimization algorithm. However, for feature selection problems, the search space is represented using binary coding [0, 1]. The position vector contains only 0 and 1, the modified update method is essential to select features that are related to each other. This is done by utilizing various transfer functions to convert continuous values into binary-based values. Transfer functions are used to provide possible changes to the position vector in binary space. Increasing/decreasing the speed at high/low probability levels for position updates is a key factor in selecting the transfer function. Therefore, choosing an appropriate transfer function by converting the continuous

search space into binary values is very important [7]. The transfer function value is obtained from the sigmoid function which will update the position of the Grey wolf. The following is the sigmoid function used in the BGWO algorithm.

$$Sigmoid(\vec{X}) = \frac{1}{1 + e^{-10(x-0.5)}} \quad (9)$$

Now, let us confirm the latest position of the Grey wolf by giving the value [0-1], and update it by using the following Eq. 10.

$$S\vec{X}(t + 1) = f(x) = \begin{cases} 1, & \text{sigmoid}\left(\frac{\vec{X}_1 + \vec{X}_2 + \vec{X}_3}{3}\right) \geq \text{rand} \\ 0, & \text{otherwise} \end{cases} \quad (10)$$

The value of the variable $\vec{X}(t + 1)$ is an additional variable used to update the position of the Grey wolf towards the food source. $Sigmoid(\vec{X})$ variable value (\vec{X}) is changed to $\frac{\vec{X}_1 + \vec{X}_2 + \vec{X}_3}{3}$ so that the value is 1 if the resulting sigmoid value is greater than the rand value and will be 0 if it is smaller or equal to the rand value. The pseudocode of the BGWO algorithm is as shown below.

BGWO Pseudocode [7]

```

1 Initialize the value of the N Grey wolf variable from the population of  $X_i (i = 1, 2, \dots, n)$ 
2 Initialize Parameter  $A, a$  and  $C$ 
3 Calculate the fitness of each search agent
4  $\vec{X}_\alpha \sim$  the best search agent
5  $\vec{X}_\beta \sim$  the second best search agent
6  $\vec{X}_\delta \sim$  the third best search agent
7 While  $t < \text{Max number of iterations}$ 
8   For each search agent
9     Update the position with the t value using the equation 6, 7, 8
10    Calculate the position of each Grey wolf equation 9, 10
11  End For
12 Update parameter BGWO ( $A, a$  and  $C$ )
13 Update  $\vec{X}_\alpha, \vec{X}_\beta, \vec{X}_\delta$ 
14  $t = t + 1$ 
15 End While

```

In the pseudocode above there is a difference between the GWO algorithm and the BGWO algorithm. The difference lies in the algorithm as in equation 10 which the GWO algorithm does not have. The equation will convert values from float or decimal to binary numbers [0-1]. This equation has been shown to enhance the fitness function value, consequently improving the accuracy of the classification algorithm.

E. SINE COSINE ALGORITHM (SCA)

SCA is a multi-agent based optimization algorithm whose role is to update rules based on the mathematical trigonometric sine and cosine functions. In general, this algorithm is very simple from a mathematical and algorithmic point of view. Similar to other meta-heuristic optimization techniques, SCA starts the search process by creating a set of search solutions/agents that are randomly positioned in the search space. Next, this solution is assessed through the use of an objective function [27]. After that the algorithm saves the best solution obtained so far, marks it as a location point and the solution is updated to produce a new solution according to the sine and cosine functions [28]. This optimization procedure will repeat continuously until it finally stops when the maximum number of iterations is reached [29]. The SCA algorithm was developed by Mirjalili and this algorithm is a method based on repeated population searches. In the field of stochastic population-based optimization methods, there are generally two types of processing, namely exploration and exploitation. The SCA algorithm makes good use of both strategies [30–32]. During the exploration phase, individual solutions are randomly combined into a single solution set in order to identify promising regions in the search space. However, in the exploitation stage, the random solution gradually changes, and the range of random changes is much smaller than that in the exploration stage. Based on these capabilities, the SCA algorithm has been widely used in various fields in machine learning [33–35]. The SCA algorithm has a population, where each agent in the search can be treated as a vector in the d-dimensional search space. The search agent in the search space will update its position with the help of stochastic equations containing sine and cosine trigonometric functions. The population in the search space is initialized randomly in the boundary search space, using equation 11 in the search agent to $X_i = (X_{i1}, X_{i2}, \dots, X_{id})$ initialized using the following equation [35]:

$$X_{ij} = X_{ij}^{lb} + \text{rand}() (X_{ij}^{ub} - X_{ij}^{lb}), j = 1: d, i = 1: Np, \quad (11)$$

where X_{ij} the j it is the dimension of the i , X_{ij}^{ub} and X_{ij}^{lb} show the value's of upper and lower bounds are defined by $i - th$ solution in the $j - th$. The $\text{rand}()$ function generates random numbers uniformly distributed in the range [0, 1], and the variable Np is the number of agents searching for the population [36]. Next, the search agent with the highest Fitness Function is considered the best search agent, then the position of the best search agent will be used as the destination point. After getting the best destination point, another search agent updates its position in the search space using the destination point as a reference. The following equation is the position update equation.

$$X_{ij}^{t+1} = X_{ij}^t + r_1 \times \sin(r_2) \times |r_3 \times P_g^t - X_{ij}^t|. \quad (12)$$

$$X_{ij}^{t+1} = X_{ij}^t + r_1 \times \cos(r_2) \times |r_3 \times P_g^t - X_{ij}^t|, \quad (13)$$

where, variable $j=1$, and $i=1: Np$ $X_i = (X_{i1}, X_{i2}, \dots, X_{id})$, variable i shows the position of the search agent with the variable t as the number of iterations. $P_g^t = (P_{g1}^t, P_{g2}^t, \dots, P_{gd}^t)$ The variable g is a search agent that has found the best fitness function and can be taken into consideration as the target position for the best value with the t value as the iteration value, so T is the maximum iteration value. The variable value b is the parameter value given by the user at the beginning and π is 3,14. Then $|\cdot|$ is the modulus operator. The r_1 value is a function of the iteration of the t value and the r_2, r_3 and r_4 values are variables with random values, while the $\text{rand}()$ function produces random values.

$$r_1 = b - b \times \left(\frac{t}{T}\right), \quad (14)$$

$$r_2 = 2 \times \pi \times \text{rand}(), \quad (15)$$

$$r_3 = 2 \times \text{rand}(), \quad (16)$$

$$r_4 = \text{rand}(), \quad (17)$$

$$X_{ij}^{t+1} = \begin{cases} X_{ij}^{t+1} = X_{ij}^t + r_1 \times \sin(r_2) \times |r_3 \times P_{gj}^t - X_{ij}^t| & \text{if } r_4 < 0.5 \\ X_{ij}^{t+1} = X_{ij}^t + r_1 \times \cos(r_2) \times |r_3 \times P_{gj}^t - X_{ij}^t| & \text{if } r_4 \geq 0.5 \end{cases} \quad (18)$$

If the value of $r_4 < 0.5$, then the sin formula will be executed, but conversely if the value of $r_4 \geq 0.5$ then the cos formula will be executed.

Sine Cosine Algorithm (SCA) Pseudocode [35]

- 1 Initialize the population $\{X_1, X_2, \dots, X_N\}$
 - 2 Initialize the parameters associated with SCA
 - 3 Calculate the objective function value for each search agent in the population
 - 4 Identify the best solution obtained so far as the destination point P_g^t
 - 5 Initialize $t = 0$, where t is iteration counter
 - 6 **While** Termination criteria is met **do**
 - 7 Calculate r_1 using Eq. (14) and generate the parameters r_2, r_3 , and r_4 randomly
 - 8 **For** each search agent **do**
 - 9 Update the position of search agents using Eq. (18)
 - 10 **End For**
 - 11 Update the current best solution (or destination point) P_g^t
 - 12 $t = t + 1$
 - 13 **End While**
 - 14 **Return** the best solution P_g^t
-

The SCA pseudocode above is the SCA algorithm used for optimization. The current SCA algorithm has undergone an algorithm change to become the Binary Sine Cosine Algorithm (BSCA). In the BSCA, a binary equation is added, namely the sigmoid algorithm as in the BGWO algorithm. And the following is the equation of the sigmoid equation.

$$\text{Sigmoid}(P_g^t) = \frac{1}{1 + \exp^{-10(P_g^t - 0.5)}}, \quad (19)$$

$$X_{ij}^{(t+1)} = \begin{cases} 1 & \text{if } \text{Sigmoid}(P_g^t) \geq 0.5 \\ 0 & \text{otherwise} \end{cases}, \quad (20)$$

$Sigmoid(p_g^t)$ is the value of the best agent variable that has found the best fitness function value, and the value of this variable will be used in determining the values 1 and 0. If the variable value $Sigmoid(p_g^t) \geq 0.5$ then the value of the variable $X_{ij}^{(t+1)}$ has a value of 1, but otherwise it would have a value of 0.

Binary Sine Cosine Algorithm (BSCA) Pseudocode [37]

- 1 Initialize the population $\{X_1, X_2, \dots, X_N\}$
- 2 Initialize the parameters associated with SCA
- 3 Calculate the objective function value for each search agent in the population
- 4 Identify the best solution obtained so far as the destination point P_g^t
- 5 Initialize $t = 0$, where t is iteration counter
- 6 **While** Termination criteria is met **do**
- 7 Calculate r_1 using Eq. (14) and generate the parameters r_2, r_3, r_4 randomly
- 8 **For** each search agent **do**
- 9 Update the position of search agents using Eq. (19)
- 10 **End For**
- 11 Convert to binary using the Eq. (20)
- 12 Update the current best solution (or destination point) P_g^t
- 13 $t = t + 1$
- 14 **End While**
- 15 **Return** the best solution P_g^t

III. PROPOSED MODEL

At this stage a model is created called the Hybrid Binary Grey Wolf Optimizer-Binary Sine Cosine Algorithm (HBGWO-BSCA). The HBGWO-BSCA is inspired by the ensemble stacking model where it combines different machine learning models to solve the same problem. The HBGWO-BSCA was developed by integrating the BGWO and BSC algorithms. In this approach, the variables from the BGWO algorithm are utilized within the BSCA process to enhance accuracy, precision, recall, and F1-Score values, which serves as a function of the model ensemble. Next, look for the parameter values in the HBGWO-BSCA for parameter N in BGWO and Alpha for the BSCA parameter. Determining parameter values will be done by entering certain values to produce the lowest fitness value which is close to 0. When the global optimum value has been obtained, then the dataset is selected for its best features using the HBGWO-BSCA, which will then produce the best value when testing the HBGWO-BSCA on the K-NN and SVM classification algorithms.

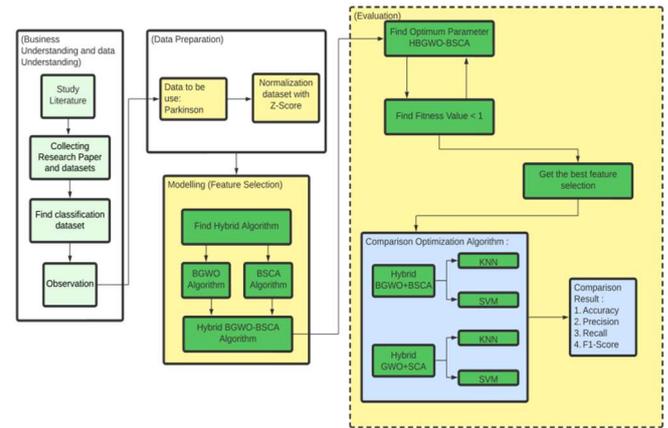


Figure 2. HBGWO-BSCA Framework

The process that must be carried out at each step of HBGWO-BSCA is Business Understanding and Data Understanding, then searching for datasets that will be later used in research. Then the existing dataset will go through a process called Z-Score normalization. The next step is modeling, where this process is the core of the research, namely the feature selection process using the HBGWO-BSCA. Then the next step is to find the parameter values for the HBGWO-BSCA to obtain the global optimum value. Determining parameter values is adjusted to the number of iteration values that have been determined up to the maximum iteration. When the iteration value reaches the maximum value, the best global optimal position will be updated. The next step is to use the selected features for testing by comparing the Hybrid Grey Wolf Optimizer-Sine Cosine Algorithm (HGWO-SCA) for feature selection with HBGWO-BSCA. This algorithm comparison uses the supervised learning algorithm K-Nearest Neighbor (KNN) and Support Vector Machine (SVM). The results of this comparison show which accuracy, precision, recall and F1-Score values are the better of the two algorithms.

IV. RESULTS

The dataset collection carried out aimed to find public datasets that could be used, such as the characteristics of the Parkinson's dataset which has 23 features. This is what makes this dataset widely used for optimization algorithms. The Parkinson's dataset is also included in the supervised learning dataset because it has already had a label or status. The Parkinson dataset has 23 features and 195 records. The Parkinson's dataset is a public dataset that has a fairly high gap value compared to several datasets in this research. This is what makes the HBGWO-BSCA a challenge that is expected to be able to overcome this problem.

A. DATA NORMALIZATION

Data normalization is used to reduce value gaps in data sets where the differences are too large. There are several normalizations that exist in data normalization, including Batch normalization (BN), Cepstral mean and variance normalization (CMVN), Weight Normalization (WN), Mean Variance Softmax Rescale (MVSr), Z-Score [38, 39]. Each normalization has its own advantages. However, in this paper

Z-Score normalization is used. Z-Score normalization is often referred to as standard score:

$$z = (X - \mu) / \sigma, \quad (21)$$

where X is the value in the dataset (the value to be normalized); μ is the average value; σ is the Standard Deviation value.

The Parkinson dataset has been normalized using the Z-Score method so that the Parkinson dataset no longer has gap values that are too far from each attribute.

B. MODEL DESIGN

Model design is a step taken to find out the model of the existing feature selection algorithm by paying attention to the following steps.

- Model Search.

In this step, we look for feature selection models such as the Grey Wolf Optimizer (GWO), Binary Grey Wolf Optimizer (BGWO), Sine Cosine Algorithm (SCA), and Binary Sine Cosine Algorithm (BSCA) feature selection models. Each of these algorithm models is usually used to solve mathematical methods, feature selection, and optimization. Writing this paper uses the HBGWO-BSCA algorithm which is used to select features in the Supervised Learning algorithm. The model search process is carried out by experimenting with combining two algorithms, namely GWO and SCA. There was research on this algorithm [42, 43]. Based on this research, the BGWO and BSC algorithms were born and combined to form a new algorithm called HBGWO-BSCA. This algorithm was created by several researches that had previously carried out experiments using the GWO and SC algorithms [44]. The GWO-SCA is mostly used to optimize functions in calculations and is not applied to perform feature selection as described in this paper [45]. These two GWO and SC algorithms were used and proven in previous papers to be successfully used for the process of solving mathematical methods or mathematical equations [46]. In this research, the feature selection process is carried out using HBGWO-BSCA on the K-NN and SVM supervised learning algorithms.

Normalization. The standard Z-Score normalization value is a data processing method [40]. Z-Score is defined as a measure of the divergence of an experimental observation in contrast to the most likely result, namely the mean. Z is expressed in the number of standard deviations from the mean value [41].

- Model Simulation

In the model simulation, an experiment is carried out using the HBGWO-BSC algorithm in selecting features. The process is as shown in Figure 3.

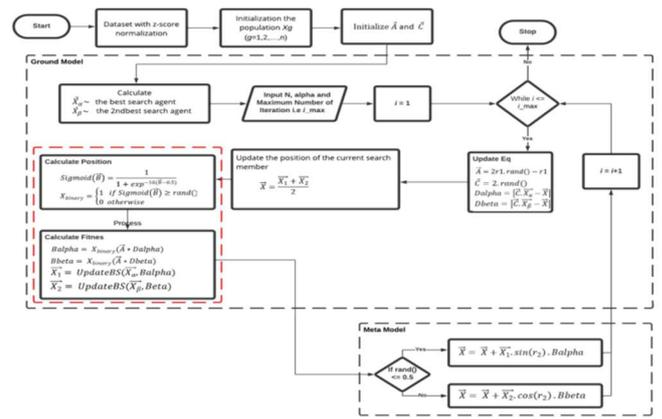


Figure 3. HBGWO-BSCA Model

In this figure, the normalized dataset then initializes the dataset population, enters parameter values in the variables, and calculates or looks for the best search agent value. Then, determine the maximum iteration value that was set. Next, the process of updating the position of the BGWO algorithm is entered into the variables \vec{X}_1 and \vec{X}_2 . After that if the rand value is smaller than 0.5, it is entered into the sine equation but otherwise, it is entered into the cosine equation. The next step will add an iteration value counter, and this process is carried out until the iteration value is maximum. And here is the pseudocode of the HBGWO-BSCA algorithm.

$$\vec{C} = 2 \cdot \text{rand}(). \quad (22)$$

The value of the variable c is a random value multiplied by 2, where the value of this variable is used to provide the positions of the wolf groups α and β in the BGWO algorithm. Then the variable values are also differentiated based on group c_1 for the α wolf group and then for c_2 which is the β group.

$$r_1 = \text{alpha} - t \cdot (\text{alpha} / t_max). \quad (23)$$

The variable r_1 will be used in variable \vec{A} where the value of r_1 is generated from the **alpha** value which must be determined manually in the HBGWO-BSCA algorithm. Then for i_max it is the maximum of iterations in the HBGWO-BSC algorithm.

$$r_2 = (2 \cdot \pi) \cdot \text{rand}(). \quad (24)$$

The r_2 variable is generated from the Phi value (3,14) multiplied by a computer random value. The result of the multiplication will be entered into the value of the variable r_2 .

$$\vec{A} = 2r_1 \cdot \text{rand}() - r_1. \quad (25)$$

Variable \vec{A} is a vector that stores the value r_1 which is multiplied by two (2) then multiplied by a random value which is reduced by the value r_1 . Then the \vec{A} value will be used as a first step in finding the optimum point in the BGWO algorithm.

$$\vec{X} = \frac{\vec{X}_1 + \vec{X}_2}{2}, \quad (26)$$

where the value of the variable \vec{X} is a vector \vec{X} . This variable is used to calculate the positioning value, namely the value of the **Dalpha** and **Dbeta**.

$$Dalpha = |\vec{C} \cdot \vec{X}_\alpha - \vec{X}|. \quad (27)$$

The **Dalpha** variable is used in the multiplier process in Eq. 29, where the **Balpha** variable is used to find binary values using the sigmoid method.

$$Dbeta = |\vec{C} \cdot \vec{X}_\beta - \vec{X}|. \quad (28)$$

The value of the **Dbeta** variable is used for the multiplier process in Eq. 30 which can also be used to find binary values using the sigmoid method.

$$Balpha = X_{binary}(\vec{A} \cdot Dalpha). \quad (29)$$

The **Balpha** variable is a function that carries out the process of selecting the values 1 and 0 in the sigmoid method for determining the BGWO and BSC algorithms, then this Balpha variable is used in \vec{X}_1 .

$$Bbeta = X_{binary}(\vec{A} \cdot Dbeta). \quad (30)$$

The **Bbeta** variable is still the same as the Alpha variable, the difference is that the results of the binary value calculation can be entered into the vector variable \vec{X}_2 .

$$Sigmoid(\vec{B}) = \frac{1}{1 + e^{-1 \cdot (\vec{B} - 0.5)}}. \quad (31)$$

This variable **Sigmoid**(\vec{B}) is generated from the function $X_{binary}(\vec{A} \cdot Dalpha)$ or $X_{binary}(\vec{A} \cdot Dbeta)$ depending on the search process for a set of wolves α and β . Then the variable \vec{B} is generated from the results of the **Balpha** and **Bbeta**.

$$X_{binary} = \begin{cases} 1 & \text{if } Sigmoid(\vec{B}) \geq rand() \\ 0 & \text{otherwise} \end{cases}. \quad (32)$$

This X_{binary} value is a function that returns values into the variables **Balpha** and **Bbeta**.

$$\vec{X}_1 = UpdateBS(\vec{X}_\alpha, Balpha). \quad (33)$$

Variable \vec{X}_1 . This is the function value of Eq. 35, to find the y value in the vector \vec{X}_1 .

$$\vec{X}_2 = UpdateBS(\vec{X}_\beta, Bbeta). \quad (34)$$

Then for the variable value \vec{X}_2 this is also the result of the function in Eq. 35 to find the y value in the vector \vec{X}_2 .

$$UpdateBS(X, B) = \begin{cases} 1 & \text{if } (X + B) \geq 1 \\ 0 & \text{otherwise} \end{cases}. \quad (35)$$

UpdateBS(X, B) function to find the **y** value used in the variable either \vec{X}_1 or \vec{X}_2 .

$$\vec{X} = \vec{X} + \vec{X}_1 \cdot \sin(r_2) \cdot Balpha. \quad (36)$$

In Eq. 36 the optimum point is determined in the HBGWO-BSC algorithm depending on the random values generated. If the random value is below 0.5 then it goes to the sine formula, but vice versa.

$$\vec{X} = \vec{X} + \vec{X}_2 \cdot \cos(r_2) \cdot Bbeta. \quad (37)$$

Equation 37 is a formula that is applied when the computer random value is above 0.5. In this case a cosine value calculation is used to determine the value. This is what makes the HBGWO-BSC algorithm very dependent on parameter selection and also computer performance in executing it step by step.

HBGWO-BSCA Pseudocode

1. Initialize population $X_g \{g = 1, 2, \dots, M\}$
 2. Initialize parameter \vec{A} and \vec{C}
 3. Calculation of the best X_α and X_β values
 4. Input parameter values N : number of Grey wolves, *alpha* : BSCA value, *i_max* : max iteration value
 5. Initialize $i = 1$, where i is the counter of the iteration
 6. **While** The criteria will stop once t_max is met **do**
 7. **For** each search agent **do**
 8. Create a random value from the BSCA variable using the equation Eq. (25),(24),(23),
 9. Update the search agent position using Eq. (27),(28),(29),(30).
 10. Update the current best solution using the equation Eq. (33),(34)
 11. Initialize $r = rand()$
 12. **if** $r \leq 0.5$
 13. use Eq. (36)
 14. **else**
 15. use Eq. (37)
 16. **end if**
 17. **End For**
 18. Convert to binary using the equation Eq. (31),(32)
 19. $i = i + 1$
 20. **End While**
 21. **Return** the best solution \vec{X}
-

Based on the pseudocode above, changes can be seen in the GWO and SC algorithms. The HBGWO-BSC algorithm is a combination of the GWO and SC algorithms that were created by several previous researchers. HBGWO-BSCA uses changes to the equation Eq. 36 and 37.

C. THE SIMULATION MODEL

The simulation model seeks for the best parameters to get a fitness value that is closest to 0. This parameter search will be tried using two parameters, including the variables N and alpha. In the Parkinson dataset, the highest parameter value is N = 10 and the alpha parameter value = 5. After obtaining a parameter value that is close to 0, the resulting features will be used in the classification algorithm. Next is to set the parameters for each SVM and KNN algorithm. For parameter settings, see Table 1.

Table 1. Parameters for the Parkinson's dataset

Parameter	SVM	KNN
Kernel Function	Gaussian	
Kernel Scale	1.56	
Box Constraint Level	3	
Multiclass method	One-vs-one	
Preset		Cosine KNN
Number of Neighbors		3
Distance Metric		City Block
Distance Weight		Equal

Parameter settings in Table 1 can produce accuracy, precision, recall and F1 Score values. The next step is to see the comparison results of the HBGWO-BSC algorithm with the HGWO-SCA on the KNN and SVM classification algorithms. Further comparison results are provided in Table 2 for more information.

Table 2. Comparison Algorithm

Dataset	Parkinson			
	Accuracy	Precision	Recall	F1-Score
HBGWO-BSCA + KNN	92.00%	90.00%	93.75%	91.84%
HBGWO-BSCA + SVM	87.50%	79.00%	95.18%	86.34%
HGWO-SCA + KNN	88.00%	81.00%	94.19%	87.10%
HGWO-SCA + SVM	88.00%	79.00%	96.34%	86.81%

Table 2 displays information on the results of the comparison between algorithms, the HBGWO-BSCA+KNN algorithm obtained the highest accuracy value of 92%. That the proposed HBGWO-BSCA+KNN algorithm outperforms the HGWO-SCA+KNN algorithm by only 88%. And the most superior precision, recall and F1 Score values are the HBGWO-BSCA+KNN algorithm. The high accuracy value in the HBGWO-BSCA+KNN algorithm is even higher because the parameter selection value is correct so it can produce a high accuracy value, besides that the feature selection resulting from the HBGWO-BSCA algorithm can produce the right features.

V. CONCLUSIONS

The results of combining the HBGWO-BSC algorithm whose features have been selected and used in the classification algorithm to find accuracy, precision, recall and F1-Score values are proven to provide the highest scores, where the accuracy value is 92%, the precision value is 90%, the recall value is 93.75% and the F1-Score value is 91.84% compared with the HGWO-SCA where the highest value for accuracy is

88%, precision value is 81%, recall value is 94.19% and F1-Score is 87.10%.

References

- [1] H. K. Bharadwaj *et al.*, "A review on the role of machine learning in enabling IoT based healthcare applications," *IEEE Access*, vol. 9, pp. 38859–38890, 2021. <https://doi.org/10.1109/ACCESS.2021.3059858>.
- [2] M. Velu *et al.*, "Human pathogenic monkeypox disease recognition using Q-Learning approach," *Diagnostics*, vol. 13, no. 8, 1491, 2023. <https://doi.org/10.3390/diagnostics13081491>.
- [3] E. J. Formeister, R. T. Baum, and J. D. Sharon, "Supervised machine learning models for classifying common causes of dizziness," *Am. J. Otolaryngol. - Head Neck Med. Surg.*, vol. 43, no. 3, p. 103402, 2022. <https://doi.org/10.1016/j.amjoto.2022.103402>.
- [4] E. Celik and S. I. Omurca, "Improving Parkinson's disease diagnosis with machine learning methods," *Proceedings of the 2019 Sci. Meet. Electr. Biomed. Eng. Comput. Sci. EBBT'2019*, 2019, pp. 1–4. <https://doi.org/10.1109/EBBT.2019.8742057>.
- [5] D. Gupta *et al.*, "Optimized cuttlefish algorithm for diagnosis of Parkinson's disease," *Cogn. Syst. Res.*, vol. 52, pp. 36–48, 2018. <https://doi.org/10.1016/j.cogsys.2018.06.006>.
- [6] M. R. Salmanpour, M. Shamsaei, and A. Rahmim, "Feature selection and machine learning methods for optimal identification and prediction of subtypes in Parkinson's disease," *Comput. Methods Programs Biomed.*, vol. 206, p. 106131, 2021. <https://doi.org/10.1016/j.cmpb.2021.106131>.
- [7] R. Ramasamy Rajammal, S. Mirjalili, G. Ekambaram, and N. Palanisamy, "Binary Grey Wolf Optimizer with mutation and adaptive k-nearest neighbour for feature selection in Parkinson's disease diagnosis," *Knowledge-Based Syst.*, vol. 246, p. 108701, 2022. <https://doi.org/10.1016/j.knosys.2022.108701>.
- [8] W. Wang, J. Lee, F. Harrou, and Y. Sun, "Early detection of Parkinson's disease using deep learning and machine learning," *IEEE Access*, vol. 8, pp. 147635–147646, 2020. <https://doi.org/10.1109/ACCESS.2020.3016062>.
- [9] M. Sivakumar, A. H. Christinal, and S. Jebasingh, "Parkinson's disease diagnosis using a combined deep learning approach," *Proceedings of the 2019 2021 3rd Int. Conf. Signal Process. Commun. ICPSC 2021*, no. May, pp. 81–84, 2021. <https://doi.org/10.1109/ICSP51351.2021.9451719>.
- [10] N. Mahendran and D. R. V. P. M., "A deep learning framework with an embedded-based feature selection approach for the early detection of the Alzheimer's disease," *Comput. Biol. Med.*, vol. 141, p. 105056, 2022. <https://doi.org/10.1016/j.combiomed.2021.105056>.
- [11] J. Hu *et al.*, "Dispersed foraging slime mould algorithm: Continuous and binary variants for global optimization and wrapper-based feature selection," *Knowledge-Based Syst.*, vol. 237, p. 107761, 2022. <https://doi.org/10.1016/j.knosys.2021.107761>.
- [12] M. S. Abbasi, H. Al-Sahaf, M. Mansoori, and I. Welch, "Behavior-based ransomware classification: A particle swarm optimization wrapper-based approach for feature selection," *Appl. Soft Comput.*, vol. 121, p. 108744, 2022. <https://doi.org/10.1016/j.asoc.2022.108744>.
- [13] Y. Guo, Z. Zhang, and F. Tang, "Feature selection with kernelized multi-class support vector machine," *Pattern Recognit.*, vol. 117, p. 107988, 2021. <https://doi.org/10.1016/j.patcog.2021.107988>.
- [14] Q. M. Alzubi, M. Anbar, Y. Sanjalawe, M. A. Al-Betar, and R. Abdullah, "Intrusion detection system based on hybridizing a modified binary grey wolf optimization and particle swarm optimization," *Expert Syst. Appl.*, vol. 204, p. 117597, 2022. <https://doi.org/10.1016/j.eswa.2022.117597>.
- [15] A. M. Ibrahim, M. A. Tawhid, and R. K. Ward, "A binary water wave optimization for feature selection," *Int. J. Approx. Reason.*, vol. 120, pp. 74–91, 2020. <https://doi.org/10.1016/j.ijar.2020.01.012>.
- [16] R. A. Ibrahim, M. Abd Elaziz, A. A. Ewees, M. El-Abd, and S. Lu, "New feature selection paradigm based on hyper-heuristic technique," *Appl. Math. Model.*, vol. 98, pp. 14–37, 2021. <https://doi.org/10.1016/j.apm.2021.04.018>.
- [17] O. M. Alyasiri, Y. N. Cheah, A. K. Abasi, and O. M. Al-Janabi, "Wrapper and hybrid feature selection methods using metaheuristic algorithms for english text classification: A systematic review," *IEEE Access*, vol. 10, pp. 39833–39852, 2022. <https://doi.org/10.1109/ACCESS.2022.3165814>.

- [18] S. Bashir, I. U. Khattak, A. Khan, F. H. Khan, A. Gani, and M. Shiraz, "A novel feature selection method for classification of medical data using filters, wrappers, and embedded approaches," *Complexity*, vol. 2022, 8190814, pp. 1-12, 2022. <https://doi.org/10.1155/2022/8190814>.
- [19] S. R. Kamel, R. Yaghoubzadeh, and M. Kheirabadi, "Improving the performance of support-vector machine by selecting the best features by Grey Wolf algorithm to increase the accuracy of diagnosis of breast cancer," *J. Big Data*, vol. 6, no. 1, pp. 90, 2019. <https://doi.org/10.1186/s40537-019-0247-7>.
- [20] M. Abdel-Basset, K. M. Sallam, R. Mohamed, I. Elgendi, K. Munasinghe, and O. M. Elkomy, "An Improved binary Grey-Wolf Optimizer with simulated annealing for feature selection," *IEEE Access*, vol. 9, pp. 139792-139822, 2021. <https://doi.org/10.1109/ACCESS.2021.3117853>.
- [21] Q. Al-Tashi, S. J. Abdul Kadir, H. M. Rais, S. Mirjalili, and H. Alhussian, "Binary optimization using hybrid Grey Wolf Optimization for feature selection," *IEEE Access*, vol. 7, pp. 39496-39508, 2019. <https://doi.org/10.1109/ACCESS.2019.2906757>.
- [22] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey Wolf Optimizer," *Adv. Eng. Softw.*, vol. 69, pp. 46-61, 2014. <https://doi.org/10.1016/j.advengsoft.2013.12.007>.
- [23] M. Shariati et al., "A novel hybrid extreme learning machine-grey wolf optimizer (ELM-GWO) model to predict compressive strength of concrete with partial replacements for cement," *Eng. Comput.*, vol. 38, no. 1, pp. 757-779, 2022. <https://doi.org/10.1007/s00366-020-01081-0>.
- [24] J. Hu et al., "Orthogonal learning covariance matrix for defects of grey wolf optimizer: Insights, balance, diversity, and feature selection," *Knowledge-Based Syst.*, vol. 213, p. 106684, 2021. <https://doi.org/10.1016/j.knsys.2020.106684>.
- [25] M. H. Hashem, H. S. Abdullah, and K. I. Ghathwan, "Grey Wolf Optimization algorithm: A surge," *Iraqi Journal of Science*, vol. 64, no. 11, pp. 5964-5984, 2023. <https://doi.org/10.24996/ijcs.2023.64.11.40>.
- [26] T. Thaher, M. Awad, M. Aldasht, A. Sheta, H. Turabieh, and H. Chantar, "An enhanced evolutionary based feature selection approach using Grey Wolf Optimizer for the classification of high-dimensional biological data," *JUCS - Journal of Universal Computer Science*, vol. 28, issue 5, pp. 499-539, 2022. <https://doi.org/10.3897/jucs.78218>.
- [27] M. H. Suid, M. A. Ahmad, M. R. T. R. Ismail, M. R. Ghazali, A. Irawan, and M. Z. Tumari, "An improved sine cosine algorithm for solving optimization problems," *Proceedings of the 2019 2018 IEEE Conf. Syst. Process Control. ICSPC'2018*, December 2018, pp. 209-213, 2018. <https://doi.org/10.1109/SPC.2018.8703982>.
- [28] B. H. Abed-ulguni, N. A. Alawad, M. A. Al-Betar, and D. Paul, "Opposition-based sine cosine optimizer utilizing refraction learning and variable neighborhood search for feature selection," *Appl. Intell.*, vol. 53, pp. 13224-13260, 2023. <https://doi.org/10.1007/s10489-022-04201-z>.
- [29] Y. Fan, S. Zhang, H. Yang, D. Xu, and Y. Wang, "An improved future search algorithm based on the sine cosine algorithm for function optimization problems," *IEEE Access*, vol. 11, pp. 30171-30187, 2023. <https://doi.org/10.1109/ACCESS.2023.3258970>.
- [30] R. Hans and H. Kaur, "Hybrid binary sine cosine algorithm and ant lion optimization (SCALO) approaches for feature selection problem," *Int. J. Comput. Mater. Sci. Eng.*, vol. 9, no. 1, pp. 1-26, 2020. <https://doi.org/10.1142/S2047684119500210>.
- [31] H. Xian, C. Yang, H. Wang, and X. Yang, "A modified sine cosine algorithm with teacher supervision learning for global optimization," *IEEE Access*, vol. 9, no. 2, pp. 17744-17766, 2021. <https://doi.org/10.1109/ACCESS.2021.3054053>.
- [32] S. Alam Ansari and A. Zafar, "A fusion of dolphin swarm optimization and improved sine cosine algorithm for automatic detection and classification of objects from surveillance videos," *Meas. J. Int. Meas. Confed.*, vol. 192, pp. 110921, 2022. <https://doi.org/10.1016/j.measurement.2022.110921>.
- [33] Y. Wan, A. Ma, L. Zhang, and Y. Zhong, "Multiobjective sine cosine algorithm for remote sensing image spatial-spectral clustering," *IEEE Trans. Cybern.*, vol. 52, no. 10, pp. 11172-11186, 2022. <https://doi.org/10.1109/TCYB.2021.3064552>.
- [34] Y. Chun and X. Hua, "Improved sine cosine algorithm for optimization problems based on self-adaptive weight and social strategy," *IEEE Access*, vol. 11, pp. 73053-73061, 2023. <https://doi.org/10.1109/ACCESS.2023.3294993>.
- [35] J. C. Bansal, P. Bajpai, A. Rawat, and A. K. Nagar, "Sine cosine algorithm for discrete optimization problems," In: *Sine Cosine Algorithm for Optimization*. Springer Briefs in Applied Sciences and Technology(). Springer, Singapore, 2023, pp. 65-86. https://doi.org/10.1007/978-981-19-9722-8_4.
- [36] P. C. Chiu, A. Selamat, O. Krejcar, and K. K. Kuok, "Hybrid sine cosine and fitness dependent optimizer for global optimization," *IEEE Access*, vol. 9, pp. 128601-128622, 2021. <https://doi.org/10.1109/ACCESS.2021.3111033>.
- [37] M. M. Eid, E. S. M. El-Kenawy, and A. Ibrahim, "A binary sine cosine-modified whale optimization algorithm for feature selection," *Proceedings of the 2019 2021 IEEE 4th Natl. Comput. Coll. Conf. NCCC 2021*, 2021, pp. 1-6. <https://doi.org/10.1109/NCCC49330.2021.9428794>.
- [38] S. Yaman, B. Karakaya, and Y. Erol, "A novel normalization algorithm to facilitate pre-assessment of Covid-19 disease by improving accuracy of CNN and its FPGA implementation," *Evol. Syst.*, vol. 14, pp. 581-591, 2023. <https://doi.org/10.1007/s12530-022-09419-3>.
- [39] K. Zhao, Z. Xu, M. Yan, T. Zhang, D. Yang, and W. Li, "A comprehensive investigation of the impact of feature selection techniques on crashing fault residence prediction models," *Inf. Softw. Technol.*, vol. 139, p. 106652, 2021. <https://doi.org/10.1016/j.infsof.2021.106652>.
- [40] Z. Gao, L. Ding, Q. Xiong, Z. Gong, and C. Xiong, "Image compressive sensing reconstruction based on z-Score standardized group sparse representation," *IEEE Access*, vol. 7, pp. 90640-90651, 2019. <https://doi.org/10.1109/ACCESS.2019.2927009>.
- [41] V. Aggarwal, V. Gupta, P. Singh, K. Sharma, and N. Sharma, "Detection of spatial outlier by using improved Z-score test," *Proceedings of the 2019 Int. Conf. Trends Electron. Informatics, ICOEI 2019*, 2019, pp. 788-790. <https://doi.org/10.1109/ICOEI.2019.8862582>.
- [42] P. C. Sahu, R. C. Prusty, and S. Panda, "Approaching hybridized GWO-SCA based type-II fuzzy controller in AGC of diverse energy source multi area power system," *J. King Saud Univ. - Eng. Sci.*, vol. 32, no. 3, pp. 186-197, 2020. <https://doi.org/10.1016/j.jksues.2019.01.004>.
- [43] N. Singh and S. B. Singh, "A novel hybrid GWO-SCA approach for optimization problems," *Eng. Sci. Technol. an Int. J.*, vol. 20, no. 6, pp. 1586-1601, 2017. <https://doi.org/10.1016/j.jestech.2017.11.001>.
- [44] W. Fu, J. Tan, X. Zhang, T. Chen, and K. Wang, "Blind parameter identification of MAR model and mutation hybrid GWO-SCA optimized SVM for fault diagnosis of rotating machinery," *Complexity*, vol. 2019, 3264969, pp. 1-17, 2019. <https://doi.org/10.1155/2019/3264969>.
- [45] R. Devarapalli, B. V. Rao, and A. Al-Durra, "Optimal parameter assessment of solar photovoltaic module equivalent circuit using a novel enhanced hybrid GWO-SCA algorithm," *Energy Reports*, vol. 8, pp. 12282-12301, 2022. <https://doi.org/10.1016/j.egyr.2022.09.069>.
- [46] W. Fu, K. Wang, C. Li, and J. Tan, "Multi-step short-term wind speed forecasting approach based on multi-scale dominant ingredient chaotic analysis, improved hybrid GWO-SCA optimization and ELM," *Energy Convers. Manag.*, vol. 187, pp. 356-377, 2019. <https://doi.org/10.1016/j.enconman.2019.02.086>.



Andi Nugroho is a lecturer in the Computer Science, Mercu Buana University, He is currently completing his doctoral studies in the Computer Science Department, Binus Graduate Program, computer science specialization, at Bina Nusantara University, Jakarta, Indonesia. His research interests are Optimization, Classification, Machine

Learning.



Harco Leslie Hendric Splits Warnars received the Ph.D. degree in computer science from Manchester Metropolitan University, Manchester, U.K. His Ph.D. thesis was on Data Mining, from 2008 to 2012. His Ph.D. was funded by the Directorate General of Higher Education, Ministry of Education and Culture, Republic of Indonesia (DIKT) Scholarship. He is currently the Head of the Concentration of Information Systems, Doctor of Computer Science (DCS), Bina Nusantara University,

and the Supervisor of the Ph.D. students in computer science. He has published more than 261 articles, which were cited by 1292 articles with Google Scholar H-index=21 and i10-index=51, as can be seen in his Google Scholar link. He has published SCOPUS, as can be seen in his SCOPUS link account with 157 documents, which were cited by 244 documents with an H-index of 12.



Sanji Muhamad Isa received the bachelor's degree from Padjadjaran University, Bandung, Indonesia, and the master's and Ph.D. degrees in computer science from the University of Indonesia. He is currently a Professor of Data Science with the Computer Science Department within the BINUS Graduate Program–Master of Computer Science. His teaching and research activities largely focus on remote sensing and biomedical engineering.



Widodo Budiharto (Senior Member, IEEE) received the bachelor's degree in physics from the University of Indonesia, Jakarta, Indonesia, the master's degree in information technology from STT Benarif, Jakarta, and the Ph.D. degree in electrical engineering from the Sepuluh Novemper Institute of Technology, Surabaya, Indonesia. He took the Ph.D. sandwich program in robotics at Kumamoto University, Japan, and conducted a postdoctoral research in robotics and artificial intelligence with Hosei University, Japan. He was an Erasmus Mundus Scholar with EU, Universite de Bourgogne, France, in 2007, and a Visiting Professor with the Erasmus Mundus French Indonesian Consortium (FICEM), France, Hosei University, in 2016 and 2017. He is currently a Professor of artificial intelligence with the School of Computer Science, Bina Nusantara University, Jakarta. His research interests include intelligent systems, data science, robotic vision, and computational intelligence.

...