IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received November 26, 2020, accepted December 22, 2020, date of publication December 30, 2020,
date of current version January 13, 2021.

Digital Object Identifier 10.1109/ACCESS.2020.3047942

A Novel Wrapper–Based Feature Selection for Early Diabetes Prediction Enhanced With a Metaheuristic

TUAN MINH LE[1], THANH MINH VO[1], TAN NHAT PHAM[2], AND SON VU TRUONG DAO[2]
[1]School of Electrical Engineering, International University, Vietnam National University Ho Chi Minh City, Ho Chi Minh City 700000, Vietnam
[2]School of Industrial Engineering and Management, International University, Vietnam National University Ho Chi Minh City, Ho Chi Minh City 700000, Vietnam

Corresponding author: Son Vu Truong Dao (dvtson@hcmiu.edu.vn)

ABSTRACT Diabetes leads to health problems for hundreds of millions of people globally every year. Available medical records of patients quantify symptoms, body features, and clinical laboratory test values, which can be used to perform biostatistics analysis aimed at finding patterns or features undetectable by current practice. In this work, we proposed a machine learning model to predict the early onset of diabetes patients. It is a novel wrapper-based feature selection utilizing Grey Wolf Optimization (GWO) and an Adaptive Particle Swam Optimization (APSO) to optimize the Multilayer Perceptron (MLP) to reduce the number of required input attributes. Moreover, we also compared the results achieved using this method and several conventional machine learning algorithms approaches such as Support Vector Machine (SVM), Decision Tree (DT), K-Nearest Neighbor (KNN), Naïve Bayesian Classifier (NBC), Random Forest Classifier (RFC), Logistic Regression (LR). Computational results of our proposed method show not only that much fewer features are needed, but also higher prediction accuracy can be achieved (96% for GWO - MLP and 97% for APGWO - MLP). This work has the potential to be applicable to clinical practice and become a supporting tool for doctors/physicians.

INDEX TERMS Neural network, early diabetes, feature selection, multilayer perceptron, grey wolf optimization.

I. INTRODUCTION

Diabetes is a chronic and worldwide disease. It is one of the eight leading causes of death in developing countries and especially in developed countries in 2012. Therefore, governments and individuals spend a large portion of their budget on researching and finding a cure for this serious illness [1]. Diabetes is a condition in which your blood sugar is consistently higher than normal due to a deficiency or insulin resistance, leading to a disorder in the metabolism of blood sugar. When patients have diabetes, their bodies are not able to break down or efficiently convert most carbohydrates they consume into sugar glucose to generate energy for daily activities. Therefore, it will cause a gradual buildup of sugar in the bloodstream. Glucose then stays in the bloodstream and does not reach every cell in the body. There are includes two types of disease: type 1 and type 2. With type 1, diabetes is that insulin deficiency occurs because the pancreas does not produce enough or not be able to produce insulin due to a congenital defect at birth. Type 1 diabetes is rare, usually occurs in young children. With type 2, patients are insulin resistant. This means that the body can still produce insulin, but it cannot metabolize glucose. About 90% of diabetes patients in the world are type 2 [2]. According to the International Diabetes Federation (IDF) [3], between the ages of 20 and 79 years, about 463 million adults are living with diabetes, by 2045 the number of people with diabetes increases to about 700 million. In the year 2019, diabetes caused at least 760 billion dollars - adults account for about 10% of total spending. Diagnosing diabetes is considered a difficult problem because it is necessary to combine different parameters to predict diabetes patients at an early stage. Nowadays, the number of patients who get diabetes is gradually increasing in the suburbs and dramatically increasing in urban areas. Therefore, the earlier for getting diagnosed, the better off it will be for the patients. However, due to the complexity of the process and other symptoms patients might have. Because of the difficulty of diagnosing early diabetes, it might cause a

The associate editor coordinating the review of this manuscript and approving it for publication was Alessandra Bertoldo.

postponement in treatment operation. Therefore, it is crucial to develop an early diabetes prediction system to support whoever works in the medical professional field to diagnose patients with more rapid and accurate conditions. Machine learning and deep learning algorithms have successfully been applied to help medical experts diagnose various diseases, such as diabetes, or heart failure. ANN has also been applied by researchers in the medical field [4], [5].

In the age of ever-evolving technology, computer technology can help diagnose diseases early and accurately, saving time and money. Many researchers applied data mining methods to diagnosing disease. This research performs an Artificial Neural Network (ANN) method with a metaheuristics-based feature selection algorithm to enhance performance. The result is a benchmark with other standard machine learning/deep learning algorithms in the following such as Logistic Regression (LR) [6], K-Nearest Neighbor Classifier (KNN) [7], Naive Bayesian Classifier (NBC) [8], Support Vector Machine (SVM) [9], Decision Tree (DT) [10], and Random Forest Classifier (RFC) [11].

According to [12], Tao Zheng proposed some of the machine learning models, including KNN, DT, NBC, Random Forest classifier (RFC), Logistic Regression (LR), Support Vector Machine (SVM) for discovering Type 2 Diabetes mellitus (T2DM) The author collected from 300 patient samples. They found accuracy 99% with LR, 96% with NB, 98% with RF, 97% with KNN, 96% with SVM, 98% with J48.

The authors in [13] used the Pima Indians Diabetes dataset available on the UCI repository. This dataset includes 768 instances and eight attributes. They applied three machine learning classification such as DT, SVM, NBC to detect diabetes patients. NBC achieves the highest accuracy of 76.30% compared to the other models.

Another method is a Machine Learning Bagging Ensemble Classifier (ML-BEC) [14]. The researcher used this method to diagnose and predict diabetes at an early stage is necessary. In the first stage of ML-BEC, it also comprises feature extraction by applying the t-distributed Stochastic technique.

On the other hand, the author in [15] used Pima Indians Diabetes dataset (738 patients). For diagnosing diabetes patients, the authors applied models such as SVM, KNN, NBC, ID3, C4.5, CART for evaluating the performance in this dataset. SVM and LDA algorithms have achieved the best accuracy at 88%.

On the contrary [16], Rabina applied supervised and unsupervised learning to diagnose diabetes patients. They used the WEKA tool to find the best classification models and the Decision Tree is better performance than the others.

In [17], Tejas N. Joshi proposed three techniques for earlier detection of diabetes patients. The author compared machine learning models including SVM, Logistic regression, and ANN with 7 attributes in the data set (Glucose, Blood Pressure, Skin Thickness, Insulin, BMI, Diabetes Pedigree Function, and age). After applying these models, the performance of the Support Vector Machine (SVM) is better than other classification methods.

In addition, M. M. Faniqul Islam using a dataset from the patients of Sylhet diabetes Hospital in Sylhet, Bangladesh with 520 instances for prediction [18]. They analyzed this dataset with some models such as NBC, LR, and RF to predict the patients, and rank the features corresponding to the most important risk factor. And the accuracy of RF has outperformed the others.

Feature selection is the process that selects a subset of relevant features to predictive modeling problems. This method can identify and remove unnecessary, irrelevant, and redundant attributes from the dataset, which does not contribute to the accuracy of the model or reduce the accuracy of the model. According to the redundancy and relevance. Yu *et al.*, [19] have classified those feature subsets into four different types: noisy and irrelevant; redundant and weakly relevant, weakly relevant and non-redundant, and powerfully relevant. An irrelevant feature does not require predicting accuracy. Furthermore, several approaches can implement with filter and wrapper methods such as models, search strategies, feature quality measures, and feature evaluation. All features play as critical and crucial factors for determining the hypothesis of the predicting models. Besides that, the number of features and size of the hypothesis spaces are directly proportional to each other, and so on. When the number of features increases, the size of the searching space also increased. One such outstanding case is that if there are M features with the binary class label in a dataset, then it has 2^{2^M} combination in the search space.

There are included three general types of Feature selection methods: wrapper method, filter method, and embedded methods.

Filter method is a selection method that is independent of the machine learning method and is a method of selection based on the relationship between the explanatory variable and the objective variable Filter methods are independent of any machine learning algorithms. So, they can be used as the input of any machine learning algorithms. There are some examples of filtering methods include the Chi-squared test, the increase in information, and the correlation coefficient score [20].

The wrapper method's performance depends on the classifier. A method of determining variable selection according to the performance of a machine learning algorithm. Try putting a subset of features into a machine learning algorithm first, and then decide whether to include features or not, depending on whether it is better or worse than the previous model (when using another feature). Some examples of the wrapper are recursive feature elimination [21], Sequential feature selection algorithms Zong [22], and genetic algorithms [23].

Thirdly, the embedded method which utilizes ensemble learning and hybrid learning methods for feature selection. It is the same as the Wrapper Method, except that it also selects variables in the machine learning algorithm simultaneously. In other words, unlike the Wrapper Method, the feature amount is determined (or weighted) at the same time as the

TABLE 1. Feature and values of the dataset.

ID	Features	Values
1	Age	[20, ….,65]
2	Sex	Male: 1, Female: 0
3	Polyuria	Yes: 1, No: 0
4	Polydipsia	Yes: 1, No: 0
5	Sudden weight loss	Yes: 1, No: 0
6	Weakness	Yes: 1, No: 0
7	Polyphagia	Yes: 1, No: 0
8	Genital blurring	Yes: 1, No: 0
9	Itching	Yes: 1, No: 0
10	Irritability	Yes: 1, No: 0
11	Delayed healing	Yes: 1, No: 0
12	Partial paresis	Yes: 1, No: 0
13	Muscle stiffness	Yes: 1, No: 0
14	Alopecia	Yes: 1, No: 0
15	Obesity	Yes: 1, No: 0
16	Class	Positive: 1, Negative: 0

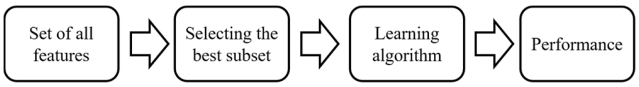

FIGURE 1. Process of filter method.

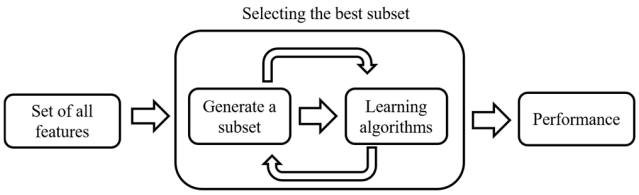

FIGURE 2. Process of a wrapper method.

model learning without combining model learning, so it is easy in terms of calculation cost (probably).

II. METHODOLOGY

A. DATA PREPROCESSING

The early-stage diabetes risk prediction dataset has been used in this research and recorded from patients using a direct questionnaire from the Sylhet Diabetes Hospital of Sylhet, Bangladesh [14]. The target of this binary classification has two categorical values: 1 – Positive and 0 – Negative. The attribute for predicting is ''Class'' which contains two categorical values and is considered as a binary classification problem. This dataset contains 520 instances and 16 attributes. Table 1 below shows the attributes and values of the dataset.

For preprocessing data, we use the IQR (Interquartile Range) method for Outlier Detection. This method is used for pre-processing data IQR in the middle spread, which is also known as the quartile range of the dataset. This concept is used in statistical analysis to help to conclude a set of numbers. IQR is used for the range of variation because it excludes most outliers of data.

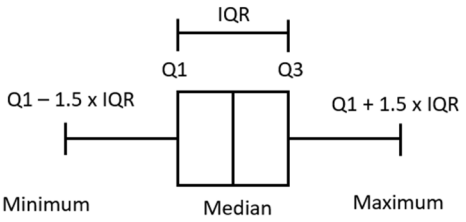

FIGURE 3. A box - Whisker plot.

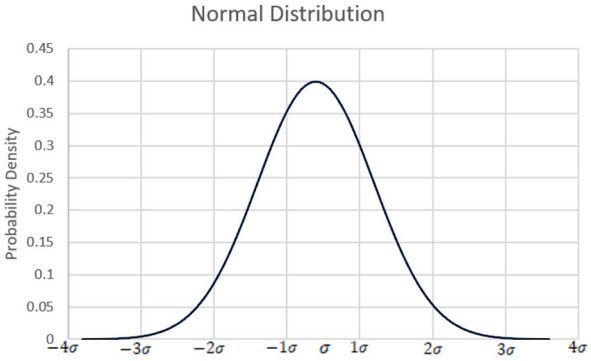

FIGURE 4. Gaussian Distribution.

In figure 3, the minimum, maximum is the minimum and maximum value in the dataset. The Median is also called the second quartile of the data. Q1 is the first quartile of the data, which means that 25% of the data lies between minimum and Q1. And Q3 is the third quartile of the data, which means that 75% of the data lies between maximum and Q3. The equation below is the Inter-Quartile Range or IQR, which is the difference between Q3 and Q1.

$$IQR = Q3 - Q1 \tag{1}$$

For detecting the outliers with this technique, we define a new range, which is called a decision range. This range is shown as given below:

$$LowerBound : (Q1 - 1.5 * IQR) \tag{2}$$

$$UpperBound : (Q3 + 1.5 * IQR) \tag{3}$$

Any data point less than the Lower Bound or more than the Upper Bound is considered as an outlier. Our data follow Normal Distribution is illustrating in figure 4.

About 68,26% of the data lies within one standard deviation (σ) of the mean (μ). About 95.44% of data lies within ($2\,\sigma$) of the mean (μ). 99.72% of data lies within (3σ) of the mean (μ). And the rest of the data lies outside (3σ) of the mean (μ). Q1 (first quartiles) and Q3 (third quartiles) lies at -0.675σ and +0.675σ from the mean (μ).

B. PREVIOUS MACHINE LEARNING MODELS

1) LOGISTIC REGRESSION (LR)

LR is an algorithm that borrows from the field of statistics [6]. This method is used for binary classification problems.

Logistic regression is the same as linear regression, whose purpose is to find out values for the coefficients. The logistics will convert any amount to about 0-1. Because of the way the model is picked up, the predictions made by logistic regression can also use as probabilities of a given data instance of class 0 or class 1. It can be useful for problems when you need to come up with multiple reasons for a prediction.

The standard logistic function σ is defined in the following:

$$\sigma(t) = \frac{1}{1+e^{t}} \tag{4}$$

The predicted output of logistic regression is usually written in the form:

$$f(x) = \sigma\left(w^{T}x\right) = \frac{1}{1+e^{-(w_0+w_1x)}} \tag{5}$$

2) K - NEAREST NEIGHBOR (KNN)

KNN is very in demand using in the Data Mining field and used to classify objects based on the closest distance between the object (Query point) and all objects in Training Data [7]. An item is classified based on its K neighbors. K is a positive integer is determined before performing the algorithm. People often use Euclidean distance to calculate the distance between objects. The Euclidean distance can be calculated in the equation below:

$$d_{Euclidean}(x,y) = \sqrt{\sum_{i=1}^{k}(x_i - y_i)^2} \tag{6}$$

KNN method is described in the following:

- Determine the K value (number of nearest neighbors)
- Calculate the distance between the classified object (Query Point) with all objects.
- Arrange the spaces in ascending order and locate the K closest neighbors to the Query Point.
- Take all classes of the nearest K identified neighbors.
- Relying on most of the closest neighbor's class to determine the class for the Query Point.

3) SUPPORT VECTOR MACHINE (SVM)

SVM is a form of the supervised machine learning model [9]. It is suitable for a relatively small data set with fewer outliers. The idea is to find a hyper lane to separate data points. This hyperplane will divide the space into different domains, and each domain will contain a type of data.

To separate the two classes of data, there are many hyperplanes, which can be chosen. Our target is finding a plane, which has the maximum margin. Margin is the distance between the hyperplane to two nearest data points corresponding to two subclasses. SVM tries to optimize the algorithm by maximizing this margin value, thereby finding the best super planar to divide the data into two layers. The data points, which are nearest to the hyperplane, are called support vectors.

A hyperplane is a linear surface, which splits the space into two sections. A hyperplane is a binary classifier and a subspace of one dimension less than its ambient space.

4) NAÏVE BAYESIAN CLASSIFIER (NBC)

NBC is an algorithm based on the Bayes theorem of probability theory to make judgments and classify data based on observed and statistical data [8]. NBC is one of the algorithms used frequently to make the most accurate predictions based on a collected dataset because it is relatively easy to understand and highly accurate. It belongs to the Supervised Machine Learning Algorithms group.

According to Bayes's theorem, the equation for calculating the random probability of y value based on the value of x:

$$P(y|x) = \frac{P(x|y)P(y)}{P(x)} \tag{7}$$

With x is a vector, which can be written as: x = (x_1, $x_2, x_3, \ldots, x_n$). Then the Bayesian equation becomes:

$$\mathrm{P(y|x1}\ldots,\mathrm{xn}) = \frac{P(x_1\,|\,y)\,P(x_2\,|\,y)\ldots P(x_n|y)P(y)}{P(x_1)\,P(x_2)\ldots P(x_n)} \tag{8}$$

Then, the target result y so that P (y | X) reaches maximum becomes:

$$y = argmax_y P(y)\prod_{i=1}^{n} p(x_i|y) \tag{9}$$

5) DECISION TREE (DT)

DT was introduced in [10]. In this method, each node of the tree will be properties, and branches are the selected value of that property. By following the property values ??on the tree, the DT will give us the predictive value. Firstly, for building a tree algorithm to predict the correct output. We analyze our data, features, and categorical (dummy values), resulting in data. Secondly, to find the best features for our tree, we should work with entropies and discriminative powers with the following formulas:

$$\begin{aligned} &Entropy{:}H(X) \\ &= \sum_{i=1}^{n} P(x_i) * \log P(x_i) \end{aligned} \tag{10}$$

$$\begin{aligned} &Discriminative\ power \\ &= entropy\,(parent) - (weight\ average) * entropy \end{aligned} \tag{11}$$

6) RANDOM FOREST CLASSIFIER (RFC)

RFC is a set of hundreds of Decision Trees. Each node of the decision tree performs a question about the data, and the branches represent possible answers to that question. Random forest is a method combining a hundred decision trees [10]. RFC models are popular due to their high accuracy and low computation costs.

RFC is a Supervised Learning method so it can handle problems of Classification and Regression.

The following steps and figure 7 describe the algorithm works:

- Select random samples from a given data set.
- Establish decision trees for each sample and get the prediction results from each decision tree.
- Vote for each prediction result.
- Choose the most predicted result as the last prediction.

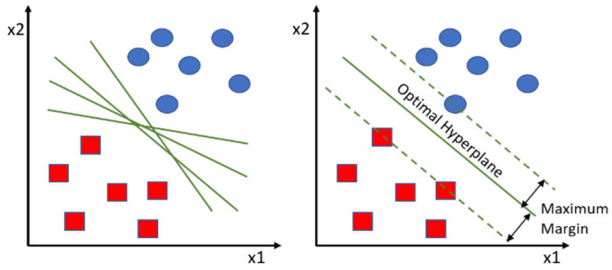

FIGURE 5. Support Vector Machine with the hyperplane.

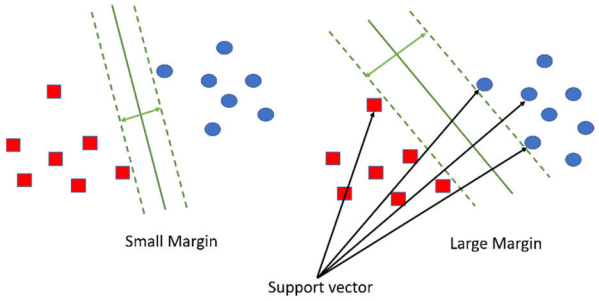

FIGURE 6. Support Vector.

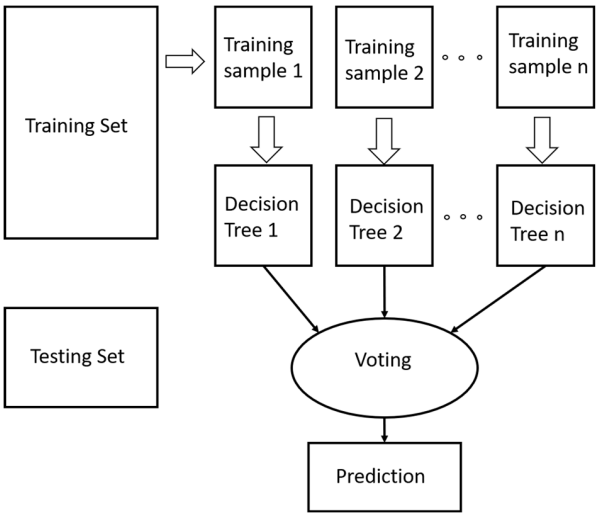

FIGURE 7. The architecture of the Random Forest Classifier.

C. PROPOSED FRAMEWORK

Metaheuristic optimization methods proposed, and many of them are a wrapper method, which has proven that this type provides better performance [24]. In [25], the author proposed a competitive binary Grey Wolf Optimizer (CBGWO) to improve the performance of BGWO [26], which is based on the Grey Wolf optimizer [27] introduced by Mirjalili for feature selection to apply to the classification of EMG (electromyography) signal. The performance of CBGWO has outperformed the other algorithms for that case study. Other wrapper feature selections were also proposed to select the best subset of features such as binary particle swarm optimization (BPSO) [28], ant colony optimization (ACO) [29], and binary differential evolution (BDE) [30]. The author in [31] proposed the Genetic Algorithms (GA) to initialize the population of BGWO. Besides that Faris *et al.* represent a review of new GWO-based performed by researchers [32].

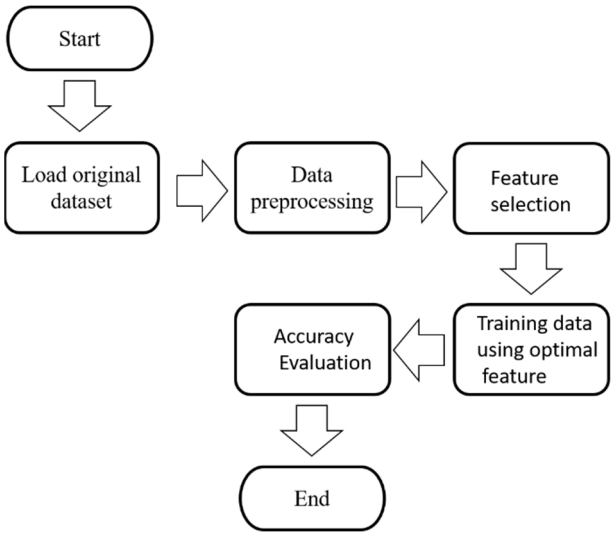

FIGURE 8. The process of our proposed method for early diabetes prediction.

In this research, after preprocessing data, we then apply feature selection which is Grey Wolf Optimize (GWO) and Adaptive Particle - Grey Wolf Optimization (APGWO) to enhance the architecture of Multilayer Perceptron (MLP) for classifying early diabetes patients. The detail of our proposed framework is described in this section. Figure 8 shows the process of our framework.

1) GREY WOLF OPTIMIZER (GWO)

Herd intelligence is the way of communication between an individual and a group. The use of herd intelligence in the fields of industry, science, and commerce has many unique and diverse applications. Research in herd intelligence can help people manage complex systems. In which GWO simulates the way that the wolves look for food and survive by avoiding their enemies. Many metaheuristic methods have been implemented to solve many real-life problems [33]–[35]. In this section, we used GWO for the optimal feature selection [27]. Figure 9 represents the hierarchy of wolves, where alphas have the leader, and omegas are the least powerful in the pack.

Alpha (α) is assumed as the leader who gives the decision for a sleeping place, hunting grey, time to wake up. The second level of gray wolves is beta (β). The betas are the wolves were typically in herds under alpha but also commanded another low-level wolf. The lowest rank among the gray wolves is Omega (Ω). Those are weak wolves and have to rely on other wolves in the pack. Delta (δ) ones are dependent on alphas and betas, but they are more effective than omega. They are responsible for monitoring territorial boundaries and

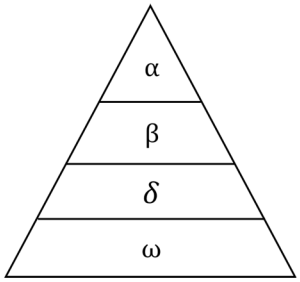

FIGURE 9. Hierarchy of wolves in GWO.

warning inside in case of danger, protect and ensure safety for herds, take care of the weak, and the ill wolves in the pack.

The inspiration of GWO is to show the way how the wolves hunting grey. The hunting process is performed in the following steps: chasing, encircling, harassing, and attacking. The following figure indicated how a wolf updates its position in the 2D search space.

For developing a mathematical model, the best solution is considered as alpha. Beta and delta are considered to the second and third solutions, respectively. The step of GWO is encircling prey is shown in the following equation.

Equation (12) represents the distance between each wolf and prey. Equation (13) is to describe the location of grey.

$$\overrightarrow{D} = \left| \overrightarrow{C} \cdot \overrightarrow{X_p}(t) - \overrightarrow{X}(t) \right| \tag{12}$$

$$\overrightarrow{X}(t+1) = \overrightarrow{X_p}(t) - \overrightarrow{A} \cdot \overrightarrow{D} \tag{13}$$

where $t$shows the current iteration, $\overrightarrow{A}$ and $\overrightarrow{C}$ are coefficient vectors, $\overrightarrow{X}$ is the position vector of a grey wolf, $\overrightarrow{X_p}$is the position vector of the prey. The coefficient $\overrightarrow{A}$ and $\overrightarrow{C}$ are indicated in the equations below:

$$\overrightarrow{A} = 2\overrightarrow{a} \cdot \overrightarrow{r}_1 - \overrightarrow{a} \tag{14}$$

$$\overrightarrow{C} = 2\overrightarrow{r}_2 \tag{15}$$

where $\overrightarrow{a}$ are linearly decreased from 2 to 0, $\overrightarrow{r}_1$ and $\overrightarrow{r}_2$ are random vector in [0, 1].

These equations below define the final position of the wolf $\overrightarrow{X}(t+1)$:

$$\overrightarrow{D}_\alpha = \left| \overrightarrow{C}_1 \cdot \overrightarrow{X_\alpha} - \overrightarrow{X} \right| \tag{16}$$

$$\overrightarrow{D}_\beta = \left| \overrightarrow{C}_2 \cdot \overrightarrow{X_\beta} - \overrightarrow{X} \right| \tag{17}$$

$$\overrightarrow{D}_\delta = \left| \overrightarrow{C}_3 \cdot \overrightarrow{X_\delta} - \overrightarrow{X} \right| \tag{18}$$

$$\overrightarrow{X}_1 = \overrightarrow{X_\alpha} - \overrightarrow{A_1} \cdot \overrightarrow{D_\alpha} \tag{19}$$

$$\overrightarrow{X}_2 = \overrightarrow{X_\beta} - \overrightarrow{A_2} \cdot \overrightarrow{D_\beta} \tag{20}$$

$$\overrightarrow{X}_3 = \overrightarrow{X_\delta} - \overrightarrow{A_3} \cdot \overrightarrow{D_\delta} \tag{21}$$

$$\overrightarrow{X}(t+1) = \frac{\overrightarrow{X}_1 + \overrightarrow{X}_2 + \overrightarrow{X}_3}{3} \tag{22}$$

In order to use for binary feature selection, the GWO updating procedure needs to be modified, as described in Emary et al.[25]. Here we utilized the BGWO2 model:

$$x_d^{t+1} = \begin{cases} 1 & if\ sigmoid\left(\frac{x_1+x_2+x_3}{3}\right) \geq rand \\ 0 & otherwise \end{cases} \tag{23}$$

The pseudo-code of GWO is given below:

For applying the feature selection method GWO, we set some parameters according to Table 2:

TABLE 2. Parameter setting of GWO.

ID	Parameter	Value
1	Max iteration	20
2	Agents	25
3	dim	Number of data
4	beta	0.01
5	alpha	1 - beta

2) ADAPTIVE PARTICLE – GREY WOLF OPTIMIZATION (APGWO)

In this research, we used a feature selection method called Adaptive Particle - Grey Wolf Optimization (APGWO) [37], which is combined from Particle Swarm Optimization [38] and Grey Wolf Optimization [27]. The Particle Swarm Optimization algorithm published by Kennedy and Eberhart in [38] and its basic judgments are mainly inspired by animals' social behavior such as birds flocking while looking for food. The birds are either scattering or walking together before they are fixing the position where they can find food. When birds are looking for food from one location to the other, there is always a bird that can smell food very well. Because they are message transmission, especially useful messages at any length of time while looking for the food. The birds will eventually fly to the position where can find food.

In these formulas below, the value of c_1 and c_2 in PSO are usually set as constants, most likely $c_1 = c_2 = 1$ or $c_1 = c_2 = 2$ to balance the exploration phases. In this research, a formula is proposed to change the coefficients of the acceleration in each iteration. The equation (24) and (25) are showing the new coefficient:

$$c_1^t = 1.2 - \frac{f\left(x_k^t\right)}{f\left(gBest\right)} \tag{24}$$

$$c_2^t = 0.5 + \frac{f\left(x_k^t\right)}{f\left(gBest\right)} \tag{25}$$

where c_1^t, c_2^t and $f\left(x_k^t\right)$ represents the coefficients and the fitness of particle k at iteration t, respectively. And $f\left(gBest\right)$ is the swarm's global best fitness. The values of 1.2 and 0.5 are also found by empirical studies. The formula for inertia is modified below:

$$w_t = (maxIter - t) * \frac{wMax - wMin}{maxIter} + wMin \tag{26}$$

Finally, the velocity and the position of the particles in the global search space have been updated in the mathematical

equations below:

$$v_k^{t+1} = w * v_k^t + c_1^t * rand * \left(pbest_k^t - x_k^t\right) + c_2^t * rand * \left(gBest - x_k^t\right) \quad (27)$$

$$x_k^{t+1} = x_k^t + v_k^t \quad (28)$$

The sigmoid function is shown in the following equation:

$$v'_{ij}(t) = sig\left(v_{ij}(t)\right) = \frac{1}{1 + e^{-v_{ij}(t)}} \quad (29)$$

The new position of the particle is obtained in (30):

$$x_{ij}(t+1) = \begin{cases} 1, & if\ r_{ij} < sig(v_{ij}(t+1)) \\ 0, & othewise \end{cases} \quad (30)$$

where *ij* is a uniform random number in the range [0,1].

Hybrid variant represents a probability of mutation, which will trigger a small number of iterations of GWO within the PSO main loop. The probability of mutation is set at 0.1 in our case. This value is set to be small so that the inner loop is only activated for a small number of times, which would not affect the quality of the solution found by the swarm as a whole.

The pseudo-code of APGWO is given below:

The solution of the APGWO-wrapper is a binary array having a dimension of 1 × n, where n is the total number of features. Selected features will take a value of 1, and 0 otherwise. The parameters set for the wrapper algorithms are as follows: 20 search agents (for PSO main loop), 20 search agents (for nested GWO loop), 50 iterations for the main PSO loop, 5 iterations for nested GWO. We set w_max = 0.9, w_min = 0.2 according to [40], where w_max and w_min are called initial weight and final weight respectively. The algorithm tries to minimize the following fitness function:

$$\textbf{Minimize}\alpha \times E_t + (1-\alpha) \times \frac{S}{L} \quad (31)$$

where E_t is the error rate on the validation set, $\alpha = 0.9$, S is the number of selected features, and L is the total number of features, which is 16 in our case. Optimizing this fitness function to improve validation accuracy as well as trying to minimize the number of selected features at the same time. The high α value means that the optimizer focuses more on improving the validation accuracy. The ANN model considered is a multi-layer perceptron (MLP) with two hidden layers, with the number of neurons being 30, and 20, respectively. Activation functions of the hidden layers are the ReLU functions. The optimizer for the MLP is Stochastic Gradient Descend (SGD) with a learning rate of 0.01, which is a common learning rate value, since a large learning rate may cause the model to converge too quickly to a local optima, while a very small value may cause a process to get stuck. The output layer contains one neuron (binary classification "0" and "1") with a sigmoid activation function. Table 3 illustrates the parameter setting of APGWO.

The APGWO is tested on 30 CEC2014 benchmark functions [41]. This set contains three unimodal functions, 13 multimodal functions, 6 hybrid functions, and 8 composition functions. These functions have many local optima, which makes them suitable for testing the metaheuristic algorithm's performance. The proposed APGWO is benchmarked with PSO and GWO. All algorithms are set to have 30 search agents and carry out 2000 iterations and they are run 20 times. The dimension of these functions is 30. The "Best" value is the best result of each function that can be found after 20 runs. The mean value and the standard deviation value indicate the accuracy and stability of the algorithm. APGWO showed competitive results compared to GWO and PSO as shown in Table A1.

TABLE 3. Parameter setting of APGWO.

ID	Parameter	Value
1	Max iteration	20
2	Agents	10
3	dim	Number of data
4	beta	0.01
5	alpha	1 - beta
6	wMax	0.9
7	wMin	0.2

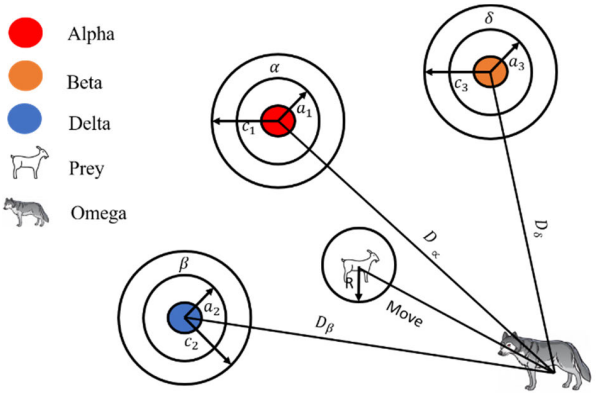

FIGURE 10. Position updating in GWO.

3) MULTILAYER PERCEPTRON (MLP)

The single-layer perceptron solves only a linearly separable problem. However, with several complex problems that are not linearly separable, one or more layers are added in a single layer perceptron, so it is known as a multilayer perceptron (MLP), which can be seen in figure 11 [40]– [43].

In the figure above, this neural network has an input layer with n neurons, one hidden layer with h neurons for each hidden layer, and an output layer with m neurons. In particular,

- Input layer: call input variable (x_1, …, x_n), also called the visible layer.
- Hidden layer: the layer of the node lies between the input and output layer.
- Output layer: this layer produces the output variables.

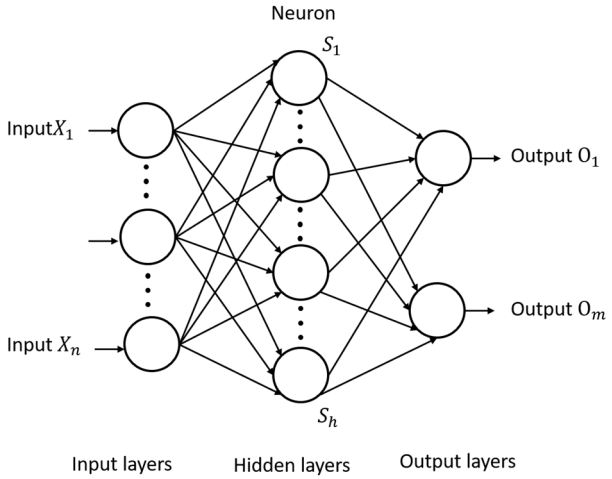

FIGURE 11. Multilayer Perceptron.

The following steps below show the calculation of the MLP output after giving the weights, inputs, and biases:

- The weighted sums of inputs are calculated as follow:

$$s_j = \sum_{i=1}^{n} (W_{ij}.X_i) - \theta_j, \quad j = 1, 2, \ldots h \tag{32}$$

where X_i shows the ith input, n: the number of nodes, W_{ij}: the connection weight from the ith node to the jth node and θ_j: the threshold of the hidden node.

- The calculation of the output of each hidden node:

$$\begin{aligned} S_j &= sigmoid\left(s_j\right) \\ &= \frac{1}{\left(1 + \exp\left(-s_j\right)\right)}, \quad j = 1, 2, \ldots h \end{aligned} \tag{33}$$

- The final outputs are based on the calculation of the output of hidden nodes:

$$o_k = \sum_{i=1}^{n} (w_{jk}.S_j) - \theta'_k, \quad k = 1, 2, \ldots m \tag{34}$$

$$\begin{aligned} O_k &= sigmoid\left(o_k\right) \\ &= \frac{1}{(1 + \exp\left(-o_k\right))}, \quad k = 1, 2, \ldots, m \end{aligned} \tag{35}$$

where w_{jk} is the connection weight from jth to kth and θ'_k is the threshold of the kth output node.

For the definition of the final output, the weights and biases are used, and we must find the values for weights and biases to achieve a relationship between the inputs and outputs. In this algorithm, weights and biases have been adjusted repeatedly for minimizing the actual output vector of the network and output vector.

III. RESULTS

This section summarizes the experimental results from applying the feature selection methods of Grey Wolf Optimization - Multilayer Perceptron (GWO) and Adaptive Particle - Grey Wolf Optimizer (APGWO) to optimize Multilayer Perceptron (MLP) and compare with the other algorithms. The next three subsections contain results and performance evaluation of the system.

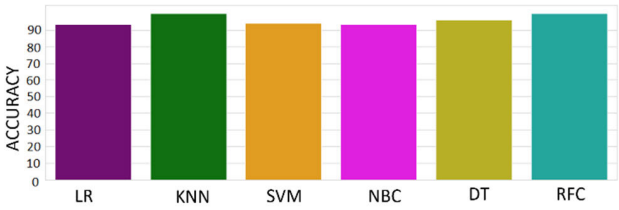

FIGURE 12. Classification accuracy of all models.

In this system, the dataset is divided into 80:20, where 80% of data for training the models, and 20% is used for testing the accuracy of these models. The performance of these algorithms is studied based on performance metrics such as accuracy, precision, recall, F_1 score, which are given in these equations below:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{36}$$

$$Precision = \frac{\text{TP}}{TP + FP} \tag{37}$$

$$Recall = \frac{\text{TP}}{TP + FN} \tag{38}$$

$$F_1 = 2 * \frac{Precision * Recall}{Precision + Recall} \tag{39}$$

where,

- True Positive (TP): the samples are classified as true (T) while they are (T).
- True Negative (TN): the samples are classified as false (F) while they are (F).
- False Positive (FP): the samples are classified as (T) while they are (F).
- False Negative (FN): the samples are classified as (F) while they are (T).

A. ANALYSIS RESULTS USING VARIOUS MACHINE LEARNING MODELS

In this study, six classifier models such as LR, KNN, SVM, NB, DT, RFC are developed. We apply the removal of the outlier dataset before training these data. The bar chart in the figure below shows the indication of the comparison of the accuracy between machine learning algorithms. As it can be seen from figure 12 that KNN and RFC have the highest accuracy with 96%. SVM and DT also achieve good accuracy compared to the others.

Figure 13 shows the confusion matrix of all algorithms. The diagonal elements of the matrix are the correctly classified number of points for each data layer. The accuracy can be inferred by the sum of those components on the diagonal divided by the sum of the elements of the entire matrix. A good model will give a confusion matrix with the factors

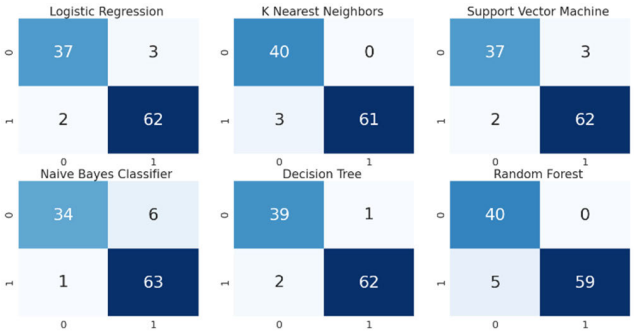

FIGURE 13. The confusion matrix of all models.

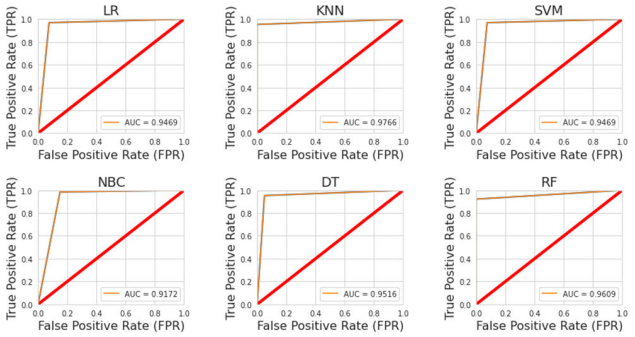

FIGURE 14. Receiver Operating Characteristic (ROC) of all models.

on the main diagonal having a big value and the remaining matters having a small value. The main purpose of diagonal elements of the matrix of KNN has a higher value than the other models.

The receiver operating characteristic (ROC) plot is a measurement for evaluating the classifier performance of each algorithm. ROC charts have been used very successfully in medical diagnosis and prognosis. A good test method will have reference points that focus on the upper left corner of the ROC chart. These points tell us they are highly sensitive and low false positives reference values. But here we have two indices (false positive and true positive), and they vary in opposite directions. The best way to normalize is to estimate the area below the ROC (also known as area under the curve - AUC). A useful test method must have an AUC area above 0.5. Figure 14 indicated that the AUC value of KNN and RF (0.9766 and 0.9609, respectively) achieved high value than the other.

B. EXPERIMENTAL RESULTS OF GWO – MLP

In the research, we show the experimental results when Grey Wolf Optimization (GWO) was applied to Multilayer Perceptron (GWO – MLP). The table below shows 13 selected features using GWO for the dataset.

Figure 15 presents the 20 iterations in terms of the fitness function value. With increasing iterations, the minimization function will decrease. We set the iteration to 20, the best fitness value is 0.038.

TABLE 4. Samples of feature selection of GWO.

ID	Features	Feature selection
1	Age	Selected
2	Sex	Selected
3	Polyuria	Selected
4	Polydipsia	Selected
5	Sudden weight loss	Selected
6	Weakness	x
7	Polyphagia	Selected
8	Genital blurring	Selected
9	Itching	Selected
10	Irritability	Selected
11	Delayed healing	Selected
12	Partial paresis	Selected
13	Muscle stiffness	Selected
14	Alopecia	Selected
15	Obesity	x

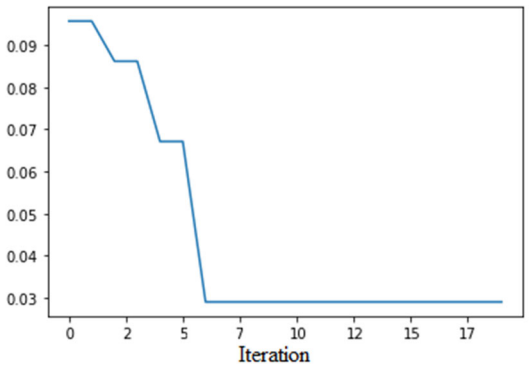

FIGURE 15. Convergence curve of the fitness function.

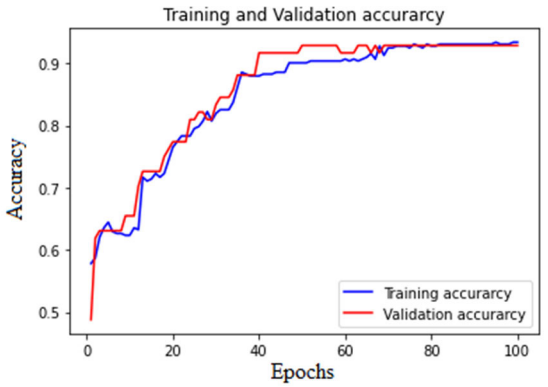

FIGURE 16. Accuracy performance of GWO - MLP.

We can see that 13 of the 16 features are selected. After that, this subset of feature is trained on the MLP with 100 epochs, which yields in figure 16 and figure 17.

In Figure 16, the APGWO-MLP method shows an accuracy of about 81% after training 200 epochs.

The loss performance of GWO research (figure 17) represents a test error that changes averagely after every 20 epochs. If it is increasing, we should stop because we are going to overfit in this case, so this stopping is called early stopping.

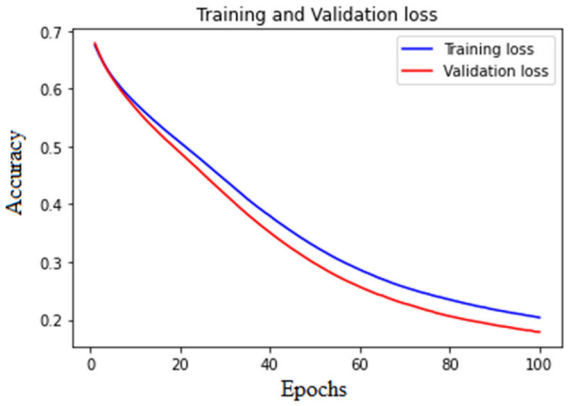

FIGURE 17. Loss performance of GWO – MLP.

TABLE 5. Samples of feature selection of APGWO.

ID	Features	Feature selection
1	Age	Selected
2	Sex	Selected
3	Polyuria	Selected
4	Polydipsia	Selected
5	Sudden weight loss	x
6	Weakness	x
7	Polyphagia	x
8	Genital blurring	x
9	Itching	Selected
10	Irritability	x
11	Delayed healing	Selected
12	Partial paresis	x
13	Muscle stiffness	Selected
14	Alopecia	Selected
15	Obesity	Selected
16	Class	Selected

The situation is that it should stop keeping weight and prevent increasing of our test error.

C. EXPERIMENTAL RESULTS OF APGWO - MLP

The following paragraph will provide information about the performance of Adaptive Particle - Grey Wolf Optimization applied to Multilayer Perceptron (APGWO - MLP). And it accomplished the selected feature in the following table.

From figure 19, with iteration from 1 to 20, the fitness value is decreased from 0.18 to 0.07.

After running the wrapper feature selection, there are 10 out of 15 features selected. Then, this subset of features is also trained on the MLP with 300 epochs, which is shown in the following results.

In figure 20, for the MLP approach with APGWO, the chart shows training accuracy and validation accuracy increase. It acquires 97% accuracy, 99% precision, 97% recall, 98% F_1 score. The proposed algorithms outperform other algorithms with the optimal feature. Moreover, the number of selected features achieved the highest classification accuracy over all

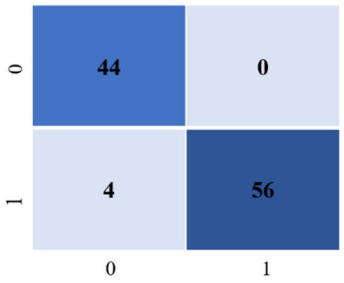

FIGURE 18. Confusion matrix of GWO – MLP.

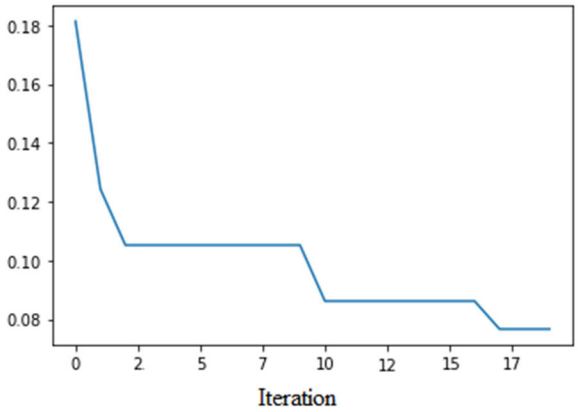

FIGURE 19. Convergence curve of the fitness function.

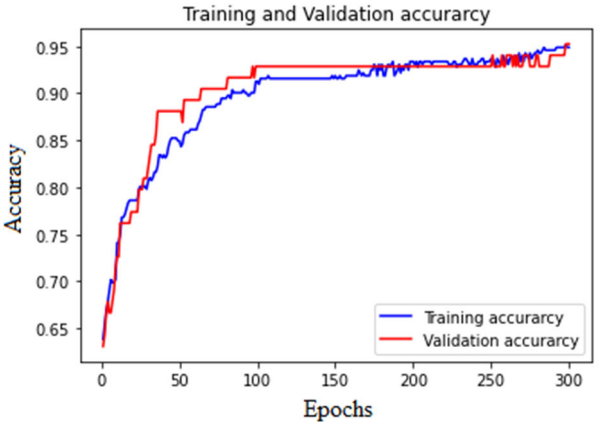

FIGURE 20. Accuracy performance of APGWO - MLP.

the others. With this approach, the training time is lower than the other models.

In the figure below, the number of epochs increases, both train and validation loss decrease. This model is suitable for practical application for diagnosing early diabetes patients.

D. PERFORMANCE EVALUATION

We apply methods several times to predict heart failure patients and reported the results in Table 6. The results indicate the comparison between the proposed method and previous machine learning models such as LR, KNN, SVM,

TABLE 6. Performance of the classification algorithms.

ID	Method	Accuracy (%)	Recall (%)	Precision (%)	F_1 Score (%)
1	LR	95.19	96.87	95.38	96.21
2	KNN	96.15	95.31	100	97.6
3	SVM	95.19	96.87	95.38	96.12
4	NBC	93.27	98.4	91.3	94.7
5	DT	95	95	97	96
6	RFC	96	95	95	95
7	GWO-MLP	96	93	100	97
8	APGWO-MLP	97	97	99	98

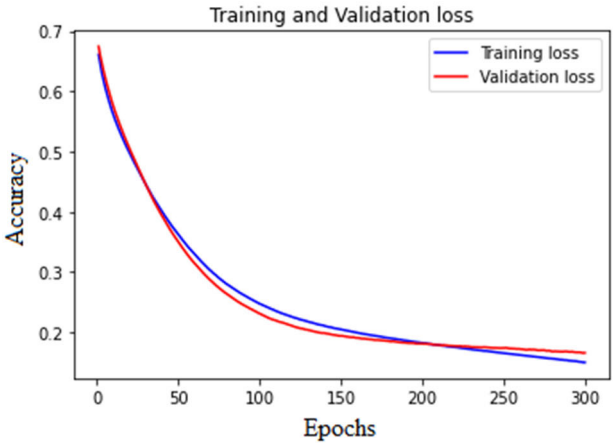

FIGURE 21. Loss performance of APGWO - MLP.

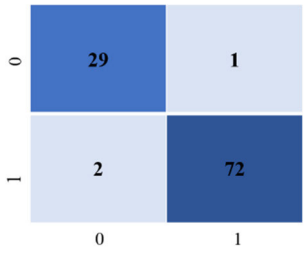

FIGURE 22. Confusion matrix of APGWO – MLP.

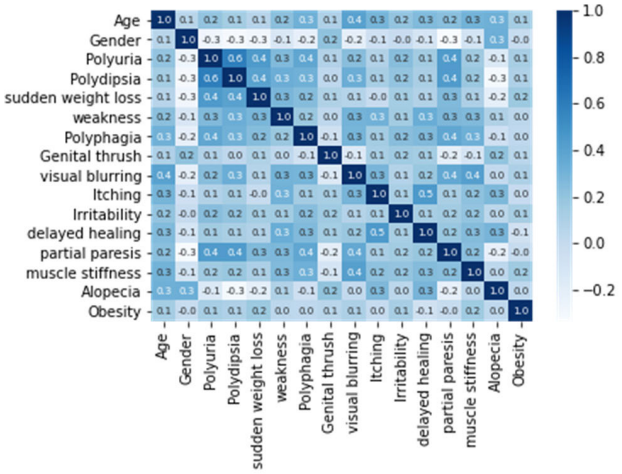

FIGURE 23. Correlation between original features.

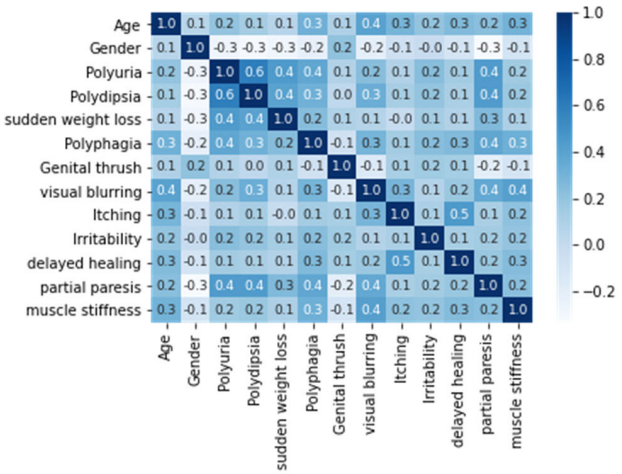

FIGURE 24. Correlation between chosen features of GWO - MLP.

NBC, DT. This comparative represents that Adaptive Particle Grey Wolf Optimization applied with Multilayer Perceptron (MLP) achieves the best accuracy.

Our prediction results (Table 6) show that APGWO-MLP outperformed the other machine learning models, by achieving the top accuracy (97%) and the top F_1score (98%).

Recall and F_1 score are also displayed in Table 6 . Besides, the NBC method gets the best recall with 98.4%. GWO-MLP and KNN get top results with a precision score (100%). While KNN, RFC, and GWO-MLP method also get higher accuracy values compare to the rest of the models. NBC gets a low performance than the other because the NBC model is its excessive simplicity. Therefore, the NBC model often does not allow interpretation of multidimensional interactions or the exploration of new pathophysiological mechanisms.

We also explore the correlation between the features, and the correlation between the fearures and the target variable. The former one helps with choosing the independent features, while the latter one helps with choosing the features that significantly affect the target variable.

Figure 23 shows the correlation between all of the features. While figure 24 illustrates the correlation of 13 selected features after using GWO.

Fig. 25 represents the heat map, which is indicated the correlation between 10 chosen features in APGWO.

Two significant features have high correlation values which are polyuria and polydipsia. Therefore, we have to perform another three tests in table VII. In case one, we remove feature polyuria. While in case two, the polydipsia feature has been eliminated. And in the last case, both two features have been discarded.

Compared with the GWO-wrapper, APGWO-wrapper can remove most of the low-correlated features. With the subset

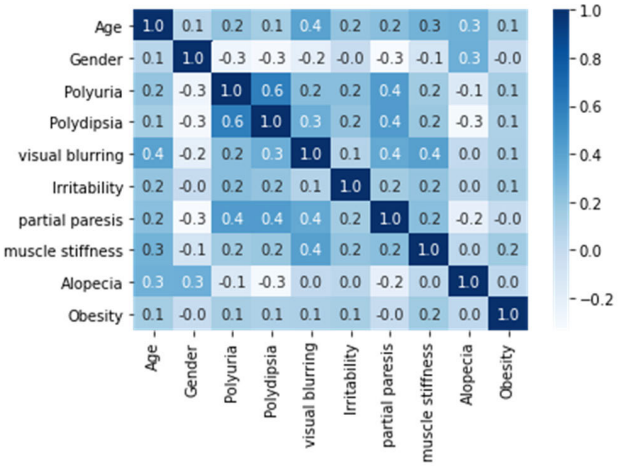

FIGURE 25. Correlation between chosen features of APGWO – MLP.

TABLE 7. Accuracy between tests.

ID	N_o OF THE FEATURES USED	GWO – MLP (%)	APGWO – MLP (%)
1	Use all features*	96	97
2	Remove polyuria**	90	92
3	Remove polydipsia***	93	93
4	Remove both****	87	92

TABLE 8. Accuracy between different activation function.

ID	HIDDEN LAYER 1	HIDDEN LAYER 2	OUTPUT LAYER	ACCURACY %
1	ReLu	ReLu	Sigmoid	97
2	ReLu	Sigmoid	Sigmoid	92
3	Sigmoid	ReLu	Sigmoid	91
4	Sigmoid	Sigmoid	Sigmoid	87
5	Sigmoid	Sigmoid	ReLu	62
6	Sigmoid	ReLu	ReLu	60
7	ReLu	Sigmoid	ReLu	97
8	ReLu	ReLu	ReLu	97

of features selected by APGWO, the overall performance of the MLP is slightly increased.

It can be seen from the table that the accuracy will be decreased when one of two features or both are removed. These two features (polyuria and polydipsia) are important because when people have diabetes, a kidney will be affected. The phenomenon of thirst can occur when people do not drink enough water, sweat a lot but do not replenish water for the body. They may also be thirsty due to diarrhea, fever, or hot weather. But for these cases, when they drink water, they will eliminate the feeling of thirst. As for thirst due to diabetes, the phenomenon of thirst continuously takes place during the day, especially at night. As soon as have finished drinking water, they may still feel thirsty and want to drink water continuously. This is because when people have diabetes, high blood sugar puts pressure on the kidneys. This will activate the kidneys to produce more urine to limit

TABLE 9. Glossary.

ACRONYMS	
ANN	Artificial Neural Network
ML	Machine Learning
KNN	K Nearét Neighbor
SVM	Support Vector Machine
LR	Logistic Regression
NBC	Naïve Bayesian Classifier
DT	Decision Tree
RFC	Random Forest Classifier
FS	Feature Selection
GWO	Grey Wolf Optimization
PSO	Particle Swarm Optimization
APGWO	Adaptive Particle Grey Wolf Optimization
MLP	Multilayer Perceptron
TP	True Positive
TN	True Negative
FP	False Positive
FN	False Negative
pbest	Personal best solution
gbest	Global best solution
AUC	Area under Curve
ROC	Receiver Operating Characteristic

excess sugar, causing dehydration and sending a signal of constant thirst.....(())[46], [47]. When people experience frequent urination, people often think they are having kidney or bladder problems. However, when they urinate more than 10 times a day, urinate more at night, and urinate more than usual, you are probably living with diabetes. With diabetes, high blood sugar causes the kidneys to work harder to reduce excess sugar. Over time, the kidneys will weaken and function less effectively. This causes excess urine, but the patient cannot control the urination. This causes tissues to become dehydrated. This is the reason why you urinate more often and feel thirsty frequently [48].

Besides, with features which were not selected such as sudden weight loss, weakness, polyphagia, genital blurring, irritability, partial paresis. In detail, sudden weight loss is maybe a phenomenon of the early sign of diabetes when people lose or gain a few pounds. Insufficient insulin prevents the body from making glucose from the bloodstream into the body's cells for use as energy. Therefore, when the body lacks insulin, the body has to start burning fat and muscle for energy to reduce the overall weight of the body. Sudden weight loss is more commonly seen in people prediagnosed with type1 diabetes [49]. In addition, polyphagia is the medical term for overeating, it could be one of the signs of diabetes. When insulin levels drop, the body's cells don't get enough glucose and the body will feel more and more hungry [50]. Moreover, irritability is a symptom genital that makes people do not care about their health. This way, people will not be able to control their blood sugar, do not have a scientific diet, make diabetes difficult to control, increase blood sugar, and dangerous complications to occur....)))[51]. On the other hand, partial paresis is the inability to control

TABLE 10. Benchmark results of APGWO, GWO, AND PSO.

Function	Actual optimum	Best			Mean			Standard Deviation		
		GWO	PSO	APGWO	GWO	PSO	APGWO	GWO	PSO	APGWO
1	100	541853.43	**4457.0149**	429898.9	8051833	**40995.91**	560747.4	6761371	**35685.89**	91832.02
2	200	5302.3764	**284.58493**	864650.77	54364509	**880.8021**	1216169	1.09E+08	**642.7583**	388493.2
3	300	841.97732	**334.59629**	2127.458	3449.915	**2050.401**	2938.121	2440.022	2778.795	**659.5567**
4	400	434.84534	**401.76255**	402.13124	436.0521	**402.4887**	403.2791	1.170979	**0.976123**	1.131843
5	500	520.35384	520.00107	**500**	520.465	520.1011	**500.845**	**0.061153**	0.064878	1.076095
6	600	601.33284	601.52137	**600.12196**	602.5458	603.6156	**600.5849**	0.765965	1.872383	**0.336801**
7	700	700.72122	700.0886	**700.08112**	701.4034	**700.1837**	700.2808	0.402896	**0.09175**	0.138014
8	800	805.03879	815.91931	**800.99496**	813.7657	826.6649	**802.3879**	7.216458	6.46033	**1.014662**
9	900	907.96623	910.94454	**902.98488**	913.4785	932.6345	**903.5819**	4.059425	14.21636	**0.795967**
10	1000	1064.0438	1608.1865	**1046.7706**	1433.25	1739.096	**1070.353**	241.6405	101.1183	**19.30522**
11	1100	1481.3056	1580.6532	**1168.135**	1701.301	1969.914	**1251.046**	185.8108	209.5975	**43.0474**
12	1200	1200.0978	1200.0303	**1200.0109**	1200.716	1200.104	**1200.052**	0.664353	0.060397	**0.032872**
13	1300	1300.1341	1300.107	**1300.0256**	1300.167	1300.225	**1300.053**	0.028156	0.075829	**0.020723**
14	1400	1400.1633	1400.2972	**1400.0843**	1400.387	1400.366	**1400.132**	0.228733	0.047151	**0.038344**
15	1500	1500.7659	**1500.2591**	1501.2338	1502.456	**1500.756**	1501.525	1.031319	0.326846	**0.234661**
16	1600	**1601.2501**	1602.1364	1601.7002	1602.571	1603.04	**1601.958**	0.730544	0.475096	**0.183149**
17	1700	2909.4626	**2213.6314**	3003.0344	126989	3377.313	**3283.403**	242617.4	1626.933	**154.111**
18	1800	2228.2805	4491.3362	**1888.1617**	9948.118	18064.67	**2019.54**	5768.639	9429.341	**105.1583**
19	1900	1901.8021	1903.0322	**1901.4933**	1902.844	1903.548	**1901.618**	0.907356	0.633904	**0.139151**
20	2000	2161.182	**2075.2267**	2082.3693	9162.69	5295.131	**2118.444**	3665.498	5088.438	**28.79687**
21	2100	6328.0984	**2236.7397**	2626.2765	8166.474	3061.773	**2734.821**	3009.836	976.9794	**101.3707**
22	2200	2238.8862	2237.0641	**2220.6132**	2272.099	2326.381	**2221.104**	43.24327	61.45893	**0.641267**
23	2300	2630.5239	2629.4575	**2300.0004**	2634.758	2629.457	**2497.676**	2.138504	**0.46523**	161.3989
24	2400	2535.0616	2567.0412	**2500**	2543.287	2587.136	**2506.439**	8.389232	15.89373	**3.842239**
25	2500	2698.9678	2670.1593	**2619.6281**	2701.162	2693.142	**2622.811**	**1.651654**	11.70806	1.893998
26	2600	2700.1397	2700.111	**2700.0359**	2720.131	2700.208	**2700.051**	39.93461	0.081562	**0.012753**
27	2700	3007.7234	3100	**2702.3648**	3053.12	3144.606	**2703.175**	32.1715	53.02392	**0.503434**
28	2800	3169.5748	3305.0835	**3050.0333**	3247.929	3714.568	**3077.562**	49.49617	370.8018	**15.98732**
29	2900	3227.3655	**3103.7644**	3176.9238	396405	**3129.471**	3209.633	785576.4	35.82135	**26.44029**
30	3000	3887.7169	4458.0453	**3557.3951**	4310.933	5047.121	**3600.961**	438.8744	417.9484	**27.21439**

sugar in the blood, it can damage blood vessels and nerves. Nerve damage that controls stimulation and physiological response can inhibit an ability to get an erection sufficient for sexual intercourse [52], [53].

E. SENSITY ANALYSIS

The activation function is a very important component of the neural network. It determines when a neuron is activated or not. An activation function is a non-linear transformation to perform an input signal. This output of this conversion will be used as neuron input on the next layer. Without activation function, weight and bias are as simple as a linear transformation function. Solving a linear function is much simpler, but it will be difficult to solve complex problems. A neural network without activation function is just a linear regression model. The Activation function supports back-propagation with the provision of errors to update the weights and bias, which makes the model self-sufficient. There are some of the popular activation functions such as Binary step, Linear, Sigmoid, Tanh, ReLu, Leaky ReLu, Softmax.

The Sigmoid function takes a real number as input and converts it to a value in the range (0; 1), which is represented in figure 26. The input is a very small negative real number, and the output is asymptotic to zero. Otherwise, if the input is a large positive real number, the output is a number asymptotic to one. The formula of the sigmoid function:

$$\sigma(\mathbf{x}) = \frac{\mathbf{1}}{\mathbf{1}+\mathbf{e}^{-\mathbf{x}}} \tag{40}$$

The ReLU function is being used a lot in recent years when training the model of a neural network. ReLU function simply filters for values ? less than zero. ReLU is proven to make the

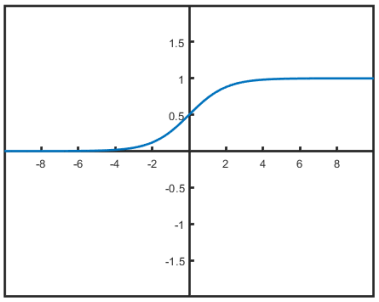

FIGURE 26. Sigmoid function.

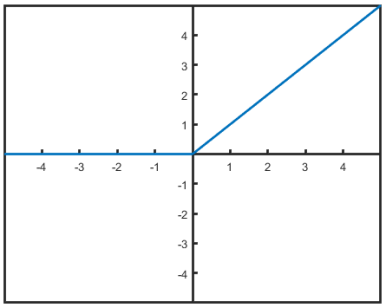

FIGURE 27. ReLu function.

training of Deep Networks much faster. The equation of the ReLu function as indicated below:

$$\mathbf{f}(\mathbf{x}) = \mathbf{max}(\mathbf{0}, \mathbf{x}) \tag{41}$$

In this research, we apply a multilayer perceptron with two hidden layers, and the number of neurons is 30 and 20, respectively. ReLu and Sigmoid are used as the activation function of the hidden layer and the output layer. Besides, we perform another test with different types of activation functions on the neural network. In each case, we run 10 times to get the average results. The following table shows the accuracy with different activation functions:

From table 8, the sigmoid function in the output layer and combination with the ReLu function in two hidden layers achieved 97% accuracy the same as when we set the Relu function in both the hidden layer and the output layer. Moreover, the accuracy unchanged when we applied the sigmoid function in the hidden layer 2. We can conclude that the sigmoid function is suitable for classification problems and the ReLU function should be used only in hidden layers.

IV. CONCLUSION

Prediction of the early onset of diabetes for patients is a difficult task due to the number of available features. Feature selection is thus being used to reduce the measurement, storage, and computation demands while maintaining high accuracy results. Considering the non-deterministic polynomial-time hard characteristic of FS, we propose a wrapper-based feature selection utilizing GWO and APGWO. We compare our results with several conventional machine learning algorithms such as SVM, DT, RFC, NBC, LR, KNN. Computational results show not only that much fewer features are needed, but also higher prediction accuracy can be achieved. In future works, we will apply an auto-tune the MLP architecture, i.e., the number of hidden nodes and hidden layers, as well as the activation functions; or optimizing the parameters of the feature selection algorithm for achieving a better performance. This work has the potential to be applicable to real-life clinical practice and become a supporting tool for doctors.

V. APPENDIX

See Tables 9 and 10.

ACKNOWLEDGMENT

The authors would like to thank M. M. Faniqul Islam, Rahatara Ferdousi, Sadikur Rahman, Humayra, and Yasmin Bushra who sharing the early-stage diabetes risk prediction dataset on the UCI Machine Learning Repository. We would also like to thank Tri Huynh, Thinh Huynh, and all graduate students in International University who gave him professional guidance and the insightful idea that considerably help him to improve the manuscript.

REFERENCES

[1] D. Falvo and B. E. Holland, *Medical and Psychosocial Aspects of Chronic Illness and Disability*. Burlington, MA, USA: Jones & Bartlett Learning, 2017.

[2] G. Klöppel, M. Löhr, K. Habich, M. Oberholzer, and P. U. Heitz, "Islet pathology and the pathogenesis of type 1 and type 2 diabetes mellitus revisited," *Pathol. Immunopathology Res.*, vol. 4, no. 2, pp. 110–125, 1985, doi: 10.1159/000156969.

[3] *International Diabetes Federation—Facts & Figures*. Accessed: Dec. 24, 2020. [Online]. Available: https://www.idf.org/aboutdiabetes/what-is-diabetes/facts-figures.html

[4] C. S. Dangare and S. S. Apte, "A data mining approach for prediction of heart disease using neural networks," *ResearchGate*, vol. 3, no. 3, pp. 30–40, 2012.

[5] S. Smiley. (Jan. 12, 2020). *Diagnostic for Heart Disease with Machine Learning*. Medium. Accessed: Sep. 19, 2020. [Online]. Available: https://towardsdatascience.com/diagnostic-for-heart-disease-with-machine-learning-81b064a3c1dd

[6] R. E. Wright, "Logistic regression," in *Reading and Understanding Multivariate Statistics*. Washington, DC, US: American Psychological Association, 1995, pp. 217–244.

[7] *An Introduction to Kernel and Nearest-Neighbor Nonparametric Regression: The American Statistician*. Accessed: Sep. 6, 2020. [Online]. Available: https://www.tandfonline.com/doi/abs/10.1080/00031305.1992.10475879

[8] K. M. Ting and Z. Zheng, "Improving the performance of boosting for naive Bayesian classification," in *Proc. Methodol. Knowl. Discovery Data Mining*, Berlin, Germany, 1999, pp. 296–305, doi: 10.1007/3-540-48912-6_41.

[9] N. V. Vapnik, *Statistical Learning Theory*. Hoboken, NJ, USA: Wiley, Sep. 1998. Accessed Sep. 6, 2020.

[10] J. R. Quinlan, "Induction of decision trees," *Mach. Learn.*, vol. 1, no. 1, pp. 81–106, Mar. 1986, doi: 10.1007/BF00116251.

[11] L. Breiman, "Random forests," *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, Oct. 2001, doi: 10.1023/A:1010933404324.

[12] T. Zheng, W. Xie, L. Xu, X. He, Y. Zhang, M. You, G. Yang, and Y. Chen, "A machine learning-based framework to identify type 2 diabetes through electronic health records," *Int. J. Med. Informat.*, vol. 97, pp. 120–127, Jan. 2017, doi: 10.1016/j.ijmedinf.2016.09.014.

[13] D. Sisodia and D. S. Sisodia, "Prediction of diabetes using classification algorithms," *Procedia Comput. Sci.*, vol. 132, pp. 1578–1585, Jan. 2018, doi: 10.1016/j.procs.2018.05.122.

[14] S. K. Somasundaram and P. Alli, "A machine learning ensemble classifier for early prediction of diabetic retinopathy," *J. Med. Syst.*, vol. 41, no. 12, p. 201, Nov. 2017, doi: 10.1007/s10916-017-0853-x.

[15] P. Agrawal and A. k. Dewangan. (2015). *A Brief Survey On The Techniques Used For The Diagnosis Of Diabetes-Mellitus*. Accessed: Nov. 10, 2020. [Online]. Available: paper/A-BRIEF-SURVEY-ON-THE-TECHNIQUES-USED-FOR-THE-OF-Agrawal-Dewangan/198dd36f4a84386817693eaa55b600005e059abd

[16] E. Rabina and E. A. Chopra. (2016). *Diabetes Prediction by Supervised and Unsupervised Learning With Feature Selection*. Accessed: Nov. 10, 2020. [Online]. Available: paper/Diabetes-Prediction-by-Supervised-and-Unsupervised-Rabina-Chopra/105093e74f7720ac14ddafa1790ca9a48f119d6e

[17] T. N. Joshi and P. N. Chawan, "Diabetes prediction using machine learning techniques," *Int. J. Eng. Res. Appl.*, vol. 8, no. 1.

[18] M. M. F. Islam, R. Ferdousi, S. Rahman, and H. Y. Bushra, "Likelihood prediction of diabetes at early stage using data mining techniques," in *Proc. Comput. Vis. Mach. Intell. Med. Image Anal.*, Singapore, 2020, pp. 113–125, doi: 10.1007/978-981-13-8798-2_12.

[19] L. Yu and H. Liu, "Efficient feature selection via analysis of relevance and redundancy," *J. Mach. Learn. Res.*, vol. 5, no. 10, pp. 1205–1224, 2004.

[20] Y. Yang and J. Pedersen, "A comparative study on feature selection in text categorization," Tech. Rep., 1997. [Online]. Available: https://dl.acm.org/doi/10.5555/645526.657137

[21] K. Yan and D. Zhang, "Feature selection and analysis on correlated gas sensor data with recursive feature elimination," *Sens. Actuators B, Chem.*, vol. 212, pp. 353–363, Jun. 2015, doi: 10.1016/j.snb.2015.02.025.

[22] A. Jain and D. Zongker, "Feature selection: Evaluation, application, and small sample performance," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 19, no. 2, pp. 153–158, Feb. 1997, doi: 10.1109/34.574797.

[23] C. De Stefano, F. Fontanella, C. Marrocco, and A. Scotto di Freca, "A GA-based feature selection approach with an application to handwritten character recognition," *Pattern Recognit. Lett.*, vol. 35, pp. 130–141, Jan. 2014, doi: 10.1016/j.patrec.2013.01.026.

[24] E. Zorarpacı and S. A. Özel, "A hybrid approach of differential evolution and artificial bee colony for feature selection," *Expert Syst. Appl.*, vol. 62, pp. 91–103, Nov. 2016, doi: 10.1016/j.eswa.2016.06.004.

[25] J. Too, A. Abdullah, N. M. Saad, N. M. Ali, and W. Tee, "A new competitive binary grey wolf optimizer to solve the feature selection problem in EMG signals classification," *Computers*, vol. 7, no. 4, p. 58, Nov. 2018, doi: 10.3390/computers7040058.

[26] E. Emary, H. M. Zawbaa, and A. E. Hassanien, "Binary grey wolf optimization approaches for feature selection," *Neurocomputing*, vol. 172, pp. 371–381, Jan. 2016, doi: 10.1016/j.neucom.2015.06.083.

[27] S. Mirjalili, S. M. Mirjalili, and A. Lewis, "Grey wolf optimizer," *Adv. Eng. Softw.*, vol. 69, pp. 46–61, Mar. 2014, doi: 10.1016/j.advengsoft.2013.12.007.

[28] L.-Y. Chuang, H.-W. Chang, C.-J. Tu, and C.-H. Yang, "Improved binary PSO for feature selection using gene expression data," *Comput. Biol. Chem.*, vol. 32, no. 1, pp. 29–38, Feb. 2008, doi: 10.1016/j.compbiolchem.2007.09.005.

[29] M. H. Aghdam, N. Ghasem-Aghaee, and M. E. Basiri, "Text feature selection using ant colony optimization," *Expert Syst. Appl.*, vol. 36, no. 3, pp. 6843–6853, Apr. 2009, doi: 10.1016/j.eswa.2008.08.022.

[30] *Feature Selection With Discrete Binary Differential Evolution—IEEE Conference Publication*. Accessed: Sep. 6, 2020. [Online]. Available: https://ieeexplore.ieee.org/document/5376334

[31] Q. Li, H. Chen, H. Huang, X. Zhao, Z. Cai, C. Tong, W. Liu, and X. Tian, "An enhanced grey wolf optimization based feature selection wrapped kernel extreme learning machine for medical diagnosis," *Comput. Math. Methods Med.*, vol. 2017, pp. 1–15, Jan. 2017. Accessed: Nov. 8, 2020. [Online]. Available: https://www.hindawi.com/journals/cmmm/2017/9512741/

[32] H. Faris, I. Aljarah, M. A. Al-Betar, and S. Mirjalili, "Grey wolf optimizer: A review of recent variants and applications," *Neural Comput. Appl.*, vol. 30, no. 2, pp. 413–435, Jul. 2018, doi: 10.1007/s00521-017-3272-5.

[33] J. Luo, H. Chen, A. A. Heidari, Y. Xu, Q. Zhang, and C. Li, "Multi-strategy boosted mutative whale-inspired optimization approaches," *Appl. Math. Model.*, vol. 73, pp. 109–123, Sep. 2019, doi: 10.1016/j.apm.2019.03.046.

[34] D. Dimitrov and H. Abdo, "Tight independent set neighborhood union condition for fractional critical deleted graphs and ID deleted graphs," *Discrete Continuous Dyn. Syst. S*, vol. 12, nos. 4–5, p. 711, 2019, doi: 10.3934/dcdss.2019045.

[35] W. Gao, H. Wu, M. K. Siddiqui, and A. Q. Baig, "Study of biological networks using graph theory," *Saudi J. Biol. Sci.*, vol. 25, no. 6, pp. 1212–1219, Sep. 2018, doi: 10.1016/j.sjbs.2017.11.022.

[36] I. Aljarah, M. Mafarja, A. A. Heidari, H. Faris, and S. Mirjalili, "Clustering analysis using a novel locality-informed grey wolf-inspired clustering approach," *Knowl. Inf. Syst.*, vol. 62, no. 2, pp. 507–539, Feb. 2020, doi: 10.1007/s10115-019-01358-x.

[37] T. N. Pham, L. V. Tran, and S. V. T. Dao, "Early disease classification of mango leaves using feed-forward neural network and hybrid Metaheuristic feature selection," *IEEE Access*, vol. 8, pp. 189960–189973, 2020, doi: 10.1109/ACCESS.2020.3031914.

[38] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proc. Int. Conf. Neural Netw., ICNN*, vol. 4, Nov. 1995, pp. 1942–1948, doi: 10.1109/ICNN.1995.488968.

[39] M. R. Girgis, A. S. Ghiduk, and E. H. Abd-Elkawy, "Automatic data flow test paths generation using the genetical swarm optimization technique," *Int. J. Comput. Appl.*, vol. 116, no. 22, pp. 25–33, Apr. 2015.

[40] D. N. Jeyakumar, T. Jayabarathi, and T. Raghunathan, "Particle swarm optimization for various types of economic dispatch problems," *Int. J. Electr. Power Energy Syst.*, vol. 28, no. 1, pp. 36–42, Jan. 2006, doi: 10.1016/j.ijepes.2005.09.004.

[41] J. J. Liang, B. Y. Qu, and P. N. Suganthan, "Problem definitions and evaluation criteria for the CEC 2014 special session and competition on single objective real-parameter numerical optimization," Zhengzhou Univ., Nanyang Technol. Univ., Singapore, Tech. Rep., 2013.

[42] M. Jahangir, H. Afzal, M. Ahmed, K. Khurshid, and R. Nawaz, "An expert system for diabetes prediction using auto tuned multi-layer perceptron," in *Proc. Intell. Syst. Conf. (IntelliSys)*, Mar. 2018, pp. 722–728.

[43] *Auto-MeDiSine: An Auto-Tunable Medical Decision Support Engine Using an Automated Class Outlier Detection Method and AutoMLP*. springerprofessional.de. Accessed: Sep. 19, 2020. [Online]. Available: https://www.springerprofessional.de/en/auto-medisine-an-auto-tunable-medical-decision-support-engine-us/16630282

[44] B. B. Chaudhuri and U. Bhattacharya, "Efficient training and improved performance of multilayer perceptron in pattern classification," *Neurocomputing*, vol. 34, no. 1, pp. 11–27, Sep. 2000, doi: 10.1016/S0925-2312(00)00305-2.

[45] U. Orhan, M. Hekim, and M. Ozer, "EEG signals classification using the K-means clustering and a multilayer perceptron neural network model," *Expert Syst. Appl.*, vol. 38, no. 10, pp. 13475–13481, Sep. 2011, doi: 10.1016/j.eswa.2011.04.149.

[46] G. L. Robertson, "Differential diagnosis of polyuria," *Annu. Rev. Med.*, vol. 39, no. 1, pp. 425–442, Feb. 1988, doi: 10.1146/annurev.me.39.020188.002233.

[47] B. P. Illowsky and D. G. Kirch, "Polydipsia and hyponatremia in psychiatric patients," *Amer. J. Psychiatry*, vol. 145, no. 6, pp. 675–683, 1988, doi: 10.1176/ajp.145.6.675.

[48] (Jun. 17, 2020). *3 P's of Diabetes: Polydipsia, Polyuria, Polyphagia, and More*. Healthline. Accessed: Nov. 18, 2020. [Online]. Available: https://www.healthline.com/health/diabetes/3-ps-of-diabetes

[49] S.-F. Weng, Y.-S. Chen, T.-C. Liu, C.-J. Hsu, and F.-Y. Tseng, "Prognostic factors of sudden sensorineural hearing loss in diabetic patients," *Diabetes Care*, vol. 27, no. 10, pp. 2560–2561, Oct. 2004, doi: 10.2337/diacare.27.10.2560-a.

[50] D. M. Atrens, "Schedule-induced polydipsia and polyphagia in nondeprived rats reinforced by intracranial stimulation," *Learn. Motivat.*, vol. 4, no. 3, pp. 320–326, Aug. 1973, doi: 10.1016/0023-9690(73)90022-2.

[51] M. J. Toohey and R. DiGiuseppe, "Defining and measuring irritability: Construct clarification and differentiation," *Clin. Psychol. Rev.*, vol. 53, pp. 93–108, Apr. 2017, doi: 10.1016/j.cpr.2017.01.009.

[52] S. Geerlings, V. Fonseca, D. Castro-Diaz, J. List, and S. Parikh, "Genital and urinary tract infections in diabetes: Impact of pharmacologically-induced glucosuria," *Diabetes Res. Clin. Pract.*, vol. 103, no. 3, pp. 373–381, Mar. 2014, doi: 10.1016/j.diabres.2013.12.052.

[53] W. A. Pulsinelli, D. E. Levy, B. Sigsbee, P. Scherer, and F. Plum, "Increased damage after ischemic stroke in patients with hyperglycemia with or without established diabetes mellitus," *Amer. J. Med.*, vol. 74, no. 4, pp. 540–544, Apr. 1983, doi: 10.1016/0002-9343(83)91007-0.

TUAN MINH LE received the bachelor's degree from the School of Electrical Engineering, International University, Vietnam National University Ho Chi Minh City, Vietnam, in 2018. He is currently following the graduated program of Electrical Engineering and is also a Laboratory Technician. His research interests include embedded systems and the Internet of Things and apply artificial intelligence.

THANH MINH VO received the B.Sc. degree from the Posts and Telecommunications Institute of Technology, Ho Chi Minh City, Vietnam, in 2004, and the M.Sc. degree in telecommunications from the Asian Institute of Technology, Bangkok, Thailand, in 2007. Since 2008, he has been a Lecturer with the School of Electrical Engineering, International University, Vietnam National University Ho Chi Minh City, Vietnam. His research interests include embedded system applications, wireless sensor networks, the Internet of Things, image processing, and embedded machine learning.

TAN NHAT PHAM received the B.Eng. degree in industrial engineering with the School of Industrial Engineering and Management, International University, Vietnam National University Ho Chi Minh City, Vietnam, in 2020. His research interests include metaheuristic algorithms and application artificial intelligence.

SON VU TRUONG DAO received the B.Eng. degree in aeronautical engineering, the M.Sc. degree in manufacturing systems and technology, and the Ph.D. degree in sensors technology, in 2004, 2005, and 2010, respectively. From 2006 to 2008, he was with Hylax Ltd., Singapore, designing high power laser systems. From 2010 to 2012, he was a Postdoctoral Fellow with Kindai University, Japan. From 2012 to 2015, he worked as a Senior Scientist with Ritsumeikan University, Japan. His research interests include development of advanced imaging sensors and systems, and applied artificial intelligence.