



Article

# Enhancement of Classifier Performance Using Swarm Intelligence in Detection of Diabetes from Pancreatic Microarray Gene Data

Dinesh Chellappan <sup>1</sup>  and Harikumar Rajaguru <sup>2,\*</sup> 

<sup>1</sup> Department of Electrical and Electronics Engineering, KPR Institute of Engineering and Technology, Coimbatore 641 407, Tamil Nadu, India; dinesh.chml@gmail.com

<sup>2</sup> Department of Electronics and Communication Engineering, Bannari Amman Institute of Technology, Sathyamangalam 638 401, Tamil Nadu, India

\* Correspondence: harikumarrajaguru@gmail.com

**Abstract:** In this study, we focused on using microarray gene data from pancreatic sources to detect diabetes mellitus. Dimensionality reduction (DR) techniques were used to reduce the dimensionally high microarray gene data. DR methods like the Bessel function, Discrete Cosine Transform (DCT), Least Squares Linear Regression (LSLR), and Artificial Algae Algorithm (AAA) are used. Subsequently, we applied meta-heuristic algorithms like the Dragonfly Optimization Algorithm (DOA) and Elephant Herding Optimization Algorithm (EHO) for feature selection. Classifiers such as Nonlinear Regression (NLR), Linear Regression (LR), Gaussian Mixture Model (GMM), Expectation Maximum (EM), Bayesian Linear Discriminant Classifier (BLDC), Logistic Regression (LoR), Softmax Discriminant Classifier (SDC), and Support Vector Machine (SVM) with three types of kernels, Linear, Polynomial, and Radial Basis Function (RBF), were utilized to detect diabetes. The classifier's performance was analyzed based on parameters like accuracy, F1 score, MCC, error rate, FM metric, and Kappa. Without feature selection, the SVM (RBF) classifier achieved a high accuracy of 90% using the AAA DR methods. The SVM (RBF) classifier using the AAA DR method for EHO feature selection outperformed the other classifiers with an accuracy of 95.714%. This improvement in the accuracy of the classifier's performance emphasizes the role of feature selection methods.



**Citation:** Chellappan, D.; Rajaguru, H. Enhancement of Classifier Performance Using Swarm Intelligence in Detection of Diabetes from Pancreatic Microarray Gene Data. *Biomimetics* **2023**, *8*, 503.

<https://doi.org/10.3390/biomimetics8060503>

Academic Editor: Huiling Chen

Received: 29 August 2023

Revised: 8 October 2023

Accepted: 20 October 2023

Published: 22 October 2023



**Copyright:** © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

**Keywords:** microarray gene data; type II DM; dimensionality reduction (DR); classification techniques; feature selection; LR; AAA; DOA; EHO

## 1. Introduction

According to the latest data from the International Diabetes Federation (IDF) Diabetes Atlas in 2021, diabetes affects around 10.5% of the global adult population aged between 20 and 79. Alarmingly, nearly half of these individuals remain unaware of their diabetes status. The projections indicate that by 2045, the number of adults living with diabetes worldwide will increase by 46% to reach approximately 783 million, which corresponds to around one in eight adults [1]. Type II DM accounts for over 90% of all diabetes cases and is influenced by several factors, including socio-economic, demographic, environmental, and genetic factors. The increase in type II DM is connected to urbanization, a growing elderly population because of higher life expectancy, reduced levels of physical activity, and a high overweight and obesity rate. To address the impact of diabetes, preventive measures, early diagnosis, and proper care for all types of diabetes are crucial. These interventions can help individuals with diabetes prevent or delay the complications associated with the condition.

According to estimates from 2019, approximately 77 million adults in India were affected by diabetes [2]. Unfortunately, the prevalence of type II DM in the country is rapidly escalating. By 2045, the number of adults living with diabetes in India could reach a staggering 134 million, with younger people under 40 being particularly affected.

Several risk factors like genetic predisposition, sedentary lifestyles, unhealthy dietary habits, obesity, urbanization, and mounting stress levels increase the risk of type II diabetes. India's southern, urban, and northern regions exhibit higher rates compared to the eastern and western regions [3]. Many cases are undiagnosed until complications arise. Diabetes continues to be the seventh leading cause of death in India, taking a toll on both human lives and the economy. It is estimated that diabetes costs the Indian economy approximately USD 100 billion annually [4].

### *Genesis of Diabetes Diagnosis Using Microarray Gene Technology*

Creating precise and effective techniques for identifying type II diabetes mellitus holds the potential to facilitate early identification and intervention. By analyzing microarray gene data, it becomes possible to identify specific genetic markers or patterns associated with diabetes [5]. This provides opportunities for personalized medicine, where treatment plans can be tailored based on an individual's genetic profile, leading to more targeted and effective interventions. Robust and reliable methods for detecting diabetes from microarray gene data can be developed and integrated into existing healthcare systems [6]. Novel dimensionality reduction techniques, classification algorithms, and feature selection methods can be explored, and other omics data can be integrated to further enhance the accuracy and reliability of diabetes detection methods. Such research could advance the state of the art in machine learning [7]. The proposed method could be used to detect other diseases that are characterized by changes in gene expression.

The structure of the article is as follows: in Section 1, an introduction to the research is discussed. Section 2 presents the literature review. Section 3 presents the methodology. In Section 4, the materials and methods are reviewed. Section 5 explains the dimensionality reduction techniques with and without a feature extraction process. In Section 6, the feature selection methods are discussed, and Section 7 focuses on the classifiers used. The results and discussion are presented in Section 8, and the conclusion is given in Section 9.

## **2. Literature Review**

Type II DM is a chronic disease that affects people worldwide irrespective of age. The early detection and diagnosis of DM in patients is essential for effective treatment and management. However, traditional methods for detecting DM, such as blood glucose testing, are often inaccurate and time consuming [8]. In recent years, there has been growing interest in the use of microarray gene data to detect DM. Microarray gene data can provide a comprehensive overview of gene expression patterns in the pancreas, which can be used to identify patients who are at risk of DM [9]. Jakka et al. [10] conducted an experimental analysis using various machine learning classifiers, including KNN, DT, NB, SVM, LR, and RF. The classifiers were trained and evaluated on the Pima Indians Diabetes dataset, which consists of nine attributes and is available from the UCI Repository. Among the classifiers tested, Logistic Regression (LR) exhibited the best performance, achieving an accuracy of 77.6%. It outperformed the other algorithms in terms of accuracy, F1 score, ROC-AUC score, and misclassification rate. Radja et al. [11] carried out a study to evaluate the performance of various supervised classification algorithms for medical data analysis, specifically in disease diagnosis. The algorithms tested included NB, SVM, decision table, and J48. The evaluation utilized measurement variables such as Correctly Classified, Incorrectly Classified, Precision, and Recall. The predictive database of diabetes was used as the testing dataset. The SVM algorithm demonstrated the highest accuracy among the tested algorithms at 77.3%, making it an effective tool for disease diagnosis. Dinh et al. [12] analyzed the capabilities of machine learning models in identifying and predicting diabetes and cardiovascular diseases using survey data, including laboratory results. The NHANES dataset was utilized, and various supervised machine learning models such as LR, SVM, RF, and GB were evaluated. An ensemble model combining the strengths of different models was developed, and key variables contributing to disease detection were identified using information obtained from tree-based models. The ensemble model achieved an AUC-ROC

score of 83.1% for cardiovascular disease detection and 86.2% for diabetes classification. When incorporating laboratory data, the accuracy increased to 83.9% for cardiovascular disease and 95.7% for diabetes. For pre-diabetic patients, the ensemble model achieved an AUC-ROC score of 73.7% without laboratory data, and XGBoost performed the best, with a score of 84.4% when using laboratory data. The key predictors for diabetes included waist size, age, self-reported weight, leg length, and sodium intake.

Yang et al. [13] conducted a study that aimed to develop prediction models for diabetes screening using an ensemble learning approach. The dataset was obtained from NHANES from 2011 to 2016. Three simple machine learning methods (LDA, SVM, and RF) were used, and the performance of the models was evaluated through fivefold cross-validation and external validation using the Delong test. The study included 8057 observations and 12 attributes. In the validation set, the ensemble model utilizing linear discriminant analysis showcased superior performance, achieving an AUC of 0.849, an accuracy of 0.730, a sensitivity of 0.819, and a specificity of 0.709. Muhammed et al. [14] conducted a study utilizing a diagnostic dataset of type 2 diabetes mellitus (DM) collected from Murtala Mohammed Specialist Hospital in Kano, Nigeria. Predictive supervised machine learning models were developed using LR, SVM, KNN, RF, NB, and GB algorithms. Among the developed models, the RF predictive learning-based model achieved the highest accuracy at 88.76%. Kim et al.'s [15] study aimed to assess the impact of nutritional intake on obesity, dyslipidemia, high blood pressure, and T2DM using deep learning techniques. The researchers developed a deep neural network (DNN) model and compared its performance with logistic regression and decision tree models. Data from the KNHANES were analyzed. The DNN model, consisting of three hidden layers with varying numbers of nodes, demonstrated superior prediction accuracy (ranging from 0.58654 to 0.80896) compared to the LoR and decision tree models. In conclusion, the study highlighted the advantage of using a DNN model over conventional machine learning models in predicting the impact of nutritional intake on obesity, dyslipidemia, high blood pressure, and T2DM.

Ramdaniah et al. [16] conducted a study utilizing microarray gene data from the GSE18732 dataset to distinguish between different classes of diabetes. The study consisted of 46 samples from diabetic classes and 72 samples from non-diabetic classes. Machine learning techniques, specifically Naïve Bayes and SVM with Sigmoid kernel, were employed for classification, achieving accuracy rates of 88.89% and 83.33%, respectively. The PIMA Indian diabetic dataset has been widely used by researchers to classify and analyze diabetic and non-diabetic patients. However, the use of microarray gene-based datasets for diabetic class identification has received less attention. As a result, a variety of performance metrics, such as accuracy, sensitivity, specificity, and MCC, have been investigated in the context of this microarray gene-based dataset.

The main characteristics and contributions of this paper are as follows:

- The work suggests a novel approach for the early detection and diagnosis of diabetes using microarray gene expression data from pancreatic sources.
- Four DR techniques are used to reduce the high dimensionality of the microarray gene data.
- Two metaheuristic algorithms are used for feature selection to further reduce the dimensionality of the microarray gene data.
- Ten classifiers in two categories, namely nonlinear models and learning-based classifiers, are used to detect diabetes mellitus. The performance of the classifiers is analyzed based on parameters like accuracy, F1 score, MCC, error rate, FM metric, and Kappa, both with and without feature selection techniques. The enhancement of classifier performance due to feature selection is exemplified through MCC and Kappa plots.

### 3. Methodology

Figure 1 shows the methodology of the research. The approach includes four DR techniques: the Bessel function (BF), Discrete Cosine Transform (DCT), Least Squares Linear Regression (LSLR), and Artificial Algae Algorithm (AAA). Following this, feature

selection techniques, either with or without classification of the data, are used. In terms of those with feature selection, two optimization algorithms are used: Elephant Herding Optimization (EHO) and Dragonfly Optimization (DFO). Moreover, ten classifiers are used, namely NLR, LR, GMM, EM, BLDC, LoR, SDC, SVM-L, SVM-Poly, and SVM-RBF, to classify the genes as non-diabetic and diabetic.

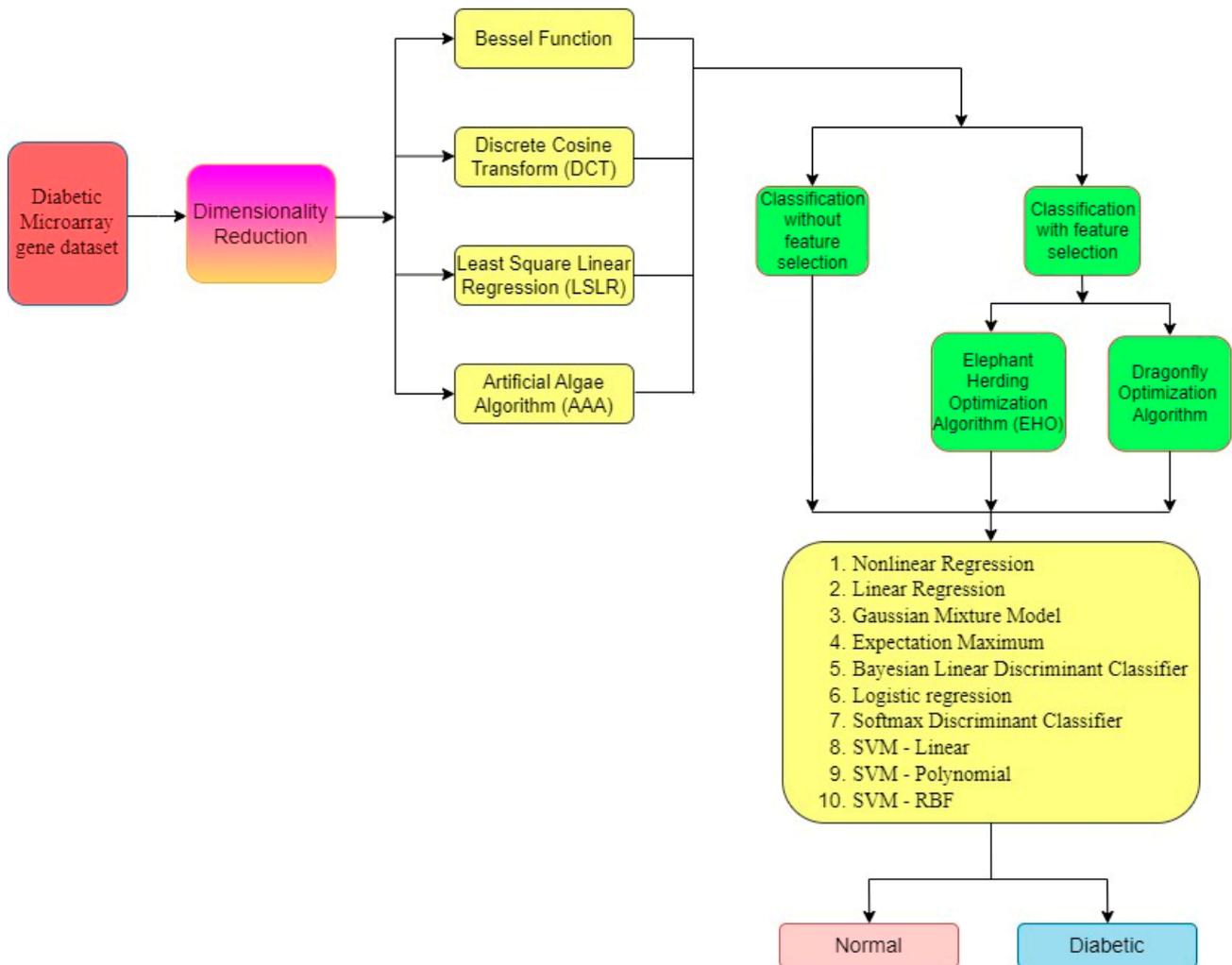


Figure 1. Workflow diagram.

Role of Microarray Gene Data

Microarray gene data play a critical role in this research. The data can be used to identify patterns of gene expression that are associated with diabetes. The data are used to train and evaluate machine learning models and to identify the most relevant features for classification. The machine learning models are then used to predict whether a patient has diabetes or not. The models are trained on a dataset of microarray gene data [17] labeled with the patient’s diabetes status.

4. Materials and Methods

Microarray gene data are readily available from many search engines. We obtained human pancreatic islet data from the Nordic Islet Transplantation program (<https://www.ncbi.nlm.nih.gov/bioproject/PRJNA178122>). The data has accessed on 20 August 2021. The dataset included 28,735 genes from 57 non-diabetic and 20 diabetic patients. The data were preprocessed to only select the 22,960 genes with the highest peak intensity per patient. The logarithmic transformation was applied with a base 10 to standardize the

individual samples, with a mean of 0 and a variance of 1. The data were then used to train and evaluate a machine learning model for the detection of diabetes. The model was able to achieve an accuracy of 90%, which is a significant improvement over the baseline accuracy of 50%. The results of this study suggest that microarray gene data can be used to develop effective methods for the detection of diabetes. The data are readily available and can be easily processed to identify the most relevant features for classification.

#### Dataset

This study focused on utilizing microarray gene data to detect diabetes and explore the features associated with the condition based on  $p$ -values using probability functions. Additionally, we aimed to address the issue of false positive errors in the selection of significant genes. The data we used for our analysis are available through multiple portals and comprised a total of 28,735 human genes as shown in the Table 1. We specifically considered 50 non-diabetic and 20 diabetic classes, selecting those with the greatest minimal intensity across 70 samples. To handle the high dimensionality of the dataset, we employed four dimensionality reduction techniques, namely BF, DCT, LSLR, and AAA. This allowed us to reduce the dimensions of the data while maintaining their informative content. The resulting dimensions were  $[2870 \times 20]$  for the diabetic group and  $[2870 \times 50]$  for the non-diabetic group. To further refine the dataset and improve classification accuracy, we applied feature selection techniques.

**Table 1.** Pancreatic microarray gene dataset for non-diabetic and diabetic classes.

Type	Total Number	Diabetic Class	Non-Diabetic Class	Total Classes
Pancreatic dataset	28,735	20	50	70

Specifically, we employed two techniques: EHO search and DOA. These techniques helped identify the most relevant features in the dataset, leading to a further reduction in dimensions to  $[287 \times 20]$  for the diabetic group and  $[287 \times 50]$  for the non-diabetic group. To evaluate the performance and accuracy of the classification, we employed ten classifiers, as already discussed.

## 5. Need for Dimensionality Reduction Techniques

Dimensionality reduction plays a crucial role in our research due to the high-dimensional nature of the microarray gene data. As the number increases, the complexity and computational costs of analyzing the data also increase significantly. Dimensionality reduction techniques allow us to reduce the number of features, making the subsequent analysis more efficient and manageable. Then, dimensionality reduction helps mitigate the curse of dimensionality [18]. In highly dimensional spaces, data points tend to become sparse, leading to difficulties in accurately representing the underlying structure of the data.

### 5.1. Dimensionality Reduction

To reduce the dimensionality of the dataset, BF, DCT, LSLR, and AAA were used.

#### Bessel Function as Dimensionality Reduction

In this section, an overview of the Bessel function and its relevant relationships and properties associated with these functions are represented [19]. Furthermore, we investigate several valuable connections and characteristics related to these functions, as  $J_n(x)$  possesses the following mathematical definition:

$$J_n(x) = \sum_{r=0}^{\infty} \frac{(-1)^r}{r! \Gamma(n+r+1)} \left(\frac{x}{2}\right)^{2r+n} \quad (1)$$

The Gamma function is represented as  $\Gamma(\lambda)$ :

$$\Gamma(\lambda) = \int_0^\infty e^{-t} t^{\lambda-1} dt \tag{2}$$

The series  $(J_n(x))$  converges for all values of  $x$  ranging from negative infinity to positive infinity. In fact, the Bessel function serves as a solution to a specific Sturm–Liouville equation [20]. This equation helps to analyze the Bessel function.

$$x^2 y''(x) + x y'(x) + (x^2 - n^2) y(x) = 0 \tag{3}$$

For  $x \in (-\infty, \infty), (n \in \mathbb{R})$ .

It is evident that the Bessel functions  $J_n(x)$  are linearly independent when  $n$  is an integer. Additionally, there exist several recursive relations for Bessel functions that can be utilized in their analysis [20]. These relations provide valuable insights into the properties and behavior of Bessel functions in various mathematical contexts.

$$\frac{d}{dx}(x^n J_n(x)) = x^n J_{n-1}(x) \tag{4}$$

$$J'_n(x) = J_{n-1}(x) - \frac{n}{x} J_n(x) \tag{5}$$

$$J'_n(x) = \frac{n}{x} J_n(x) - J_{n+1}(x) \tag{6}$$

**Lemma 1.** A significant recursion relation that proves useful in the analysis of the Bessel function of the first kind is:

$$J'_n(x) = \frac{1}{2} J_{n-1}(x) - \frac{1}{2} J_{n+1}(x) \tag{7}$$

The Bessel functions can be derived using the following procedure: Consider the vector  $J_n = [J_0(x), J_1(x), J_2(x), \dots, J_n(x)]^T$ , where  $J_0, J_1, J_2, \dots, J_n$  denote the Bessel functions evaluated at  $x$ . To obtain the derivative operational matrix, we start with the derivative of  $J_0(x)$  and denote it as  $J'_0(x)$ , where  $J'_0(x)$  represents the derivative of  $J_0(x)$  with respect to  $x$ . By constructing a matrix  $D$ , known as the derivative operational matrix, we can express  $J_n = DJ_0$ , where  $D$  is a matrix that performs the differentiation operation on  $J_0(x)$  to obtain  $J_n(x)$ . This recursion relation allows for the efficient calculation and evaluation of Bessel functions, providing a valuable tool in various mathematical and scientific applications.

$$D = \begin{bmatrix} 0 & -1 & 0 & 0 & 0 & \dots & \dots & 0 \\ 1/2 & 0 & -1/2 & 0 & 0 & \dots & \dots & 0 \\ 0 & 1/2 & 0 & -1/2 & 0 & \dots & \dots & 0 \\ 0 & 0 & 1/2 & 0 & -1/2 & \dots & \dots & 0 \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \dots & 0 \\ a_0 & a_1 & a_2 & a_3 & \dots & \dots & 0 & a_n \end{bmatrix} \tag{8}$$

DCT—Discrete Cosine Transform

The Discrete Cosine Transform (DCT) is a DR technique that approximates the Kernighan–Lin method. It aims to reduce the dimensions of the input data by eliminating the most significant features, thereby simplifying further analysis. By applying the DCT method [21], the input vector and its components are orthogonalized, resulting in a reduction in complexity. This method extracts features by selecting coefficients, which is a

crucial step with a significant impact on computation efficiency [22,23]. The DCT can be denoted as:

$$k(x) = \alpha(x) \cdot \sum_{u=0}^{s-1} a(u) \cos \frac{\pi(2u+1)x}{2s} \tag{9}$$

### Least Squares Linear Regression (LSLR) as Dimensionality Reduction

Another effective technique for reducing dimensionality is the LSLR. Hotelling [24] initially introduced this concept, utilizing principal component analysis (PCA) as a regression analysis tool. It uses principal component analysis to reduce the dimensionality of high-dimensional data before applying a linear regression model. The transformation is learned by minimizing the sum of squared errors between the predicted lower-dimensional representation and the actual high-dimensional data.

LSLR, as discussed in Hastie et al. [25], performs dimensionality reduction by identifying the best-fit line that represents the relationship between the features of independent variables and the target as a dependent variable. The objective of LSLR is to minimize the sum of squared differences between the actual and predicted values of the target variable. Considering a set of  $N$  observations in the form  $(x_1, y_1), (x_2, y_2), \dots, (x_N, y_N)$ , where  $x_i$  represents the  $i$ th observation of the independent variables and  $y_i$  corresponds to the observation of the target variable, the LSR solution can be represented as a linear equation:

$$z = \alpha_0 + \alpha_1x_1 + \alpha_2x_2 + \dots + \alpha_px_p \tag{10}$$

In the context of LSR, the linear model is characterized by the parameters  $\alpha_1, \alpha_2, \alpha_3, \dots, \alpha_p$ , where  $p$  represents the number of independent variables. This minimization process is expressed through the following equation:

$$SSE = \sum_{j=1}^m (z_i - (\alpha_0 + \alpha_1x_1 + \alpha_2x_2 + \dots + \alpha_px_p))^2 \tag{11}$$

After applying dimensionality reduction techniques to the microarray gene data, the resulting outputs are further analyzed using various statistical parameters such as mean, kurtosis, variance, Pearson correlation coefficient (PCC), skewness,  $t$ -test,  $f$ -test,  $p$ -value, and canonical correlation analysis (CCA). These statistical measures are used to assess whether the outcomes accurately represent the intrinsic properties of the underlying microarray genes in the reduced subspace.

### Artificial Algae Algorithm (AAA) as Dimensionality Reduction

The Artificial Algae Algorithm (AAA) [26] is a nature-inspired optimization algorithm that mimics the behavior and characteristics of real algae to solve complex problems. Each solution in the problem space is represented by an artificial alga, which captures the essence of algae's traits. Like real algae, artificial algae exhibit helical swimming patterns and can move towards a light source for photosynthesis. The AAA consists of three fundamental components: the evolutionary process, adaptation, and helical movement as depicted in the Figure 2. The algal colony acts as a cohesive unit, moving and responding to environmental conditions. By incorporating the principles of artificial algae into the algorithm, the AAA offers a novel approach to solving optimization problems.

$$Population = \begin{bmatrix} X_{11} & X_{12} & \dots & X_{1D} \\ X_{21} & X_{22} & \dots & X_{2D} \\ \vdots & \vdots & \ddots & \vdots \\ X_{n1} & X_{n2} & \dots & X_{nD} \end{bmatrix} \tag{12}$$

where  $X_{nD}$  is an algal cell in the  $D$ th dimension of the  $n$ th algal colony.

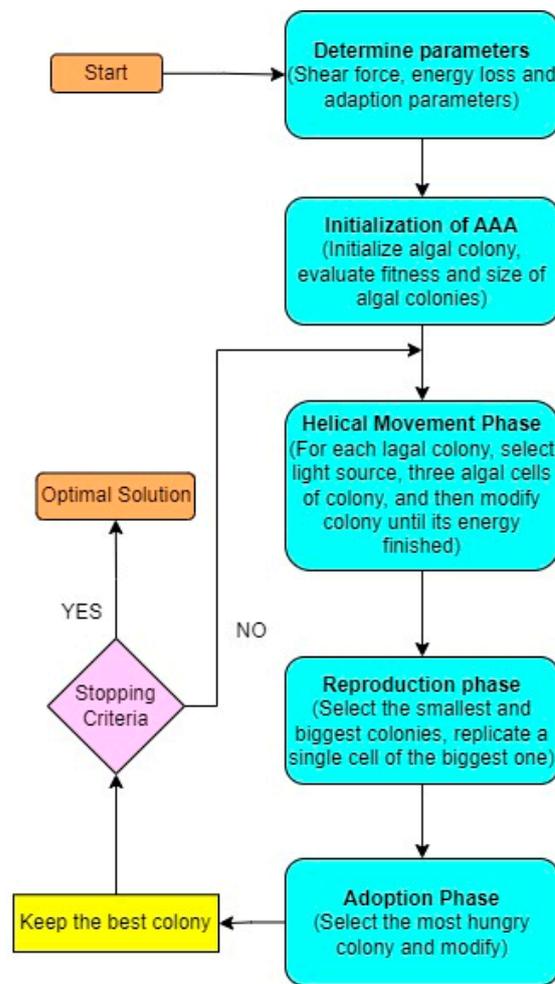


Figure 2. Flow diagram for Artificial Algae Algorithm.

During the evolutionary process [27] of the AAA, the growth and reproduction of algal colonies are influenced by the availability of nutrients and light. When an algal colony is exposed to sufficient light and nutrient conditions, it undergoes growth and replicates itself through a process like real mitotic division. In this process, two new algal cells are generated at time  $t$ . Conversely, if an algal colony does not receive enough light, it can survive for a certain period, but eventually perishes. It is important to note that  $\mu_{max}$  is assumed to be 1, as the maximum biomass conversion should be equivalent to the substrate consumption in unit time, following the conservation of mass principle. The size of the  $i$ th algal colony at time  $t + 1$  is determined by the Monod equation, as expressed in the subsequent equation:

$$H_i^t = \mu_i^t H_i^t \tag{13}$$

where  $i = 1, 2, 3, \dots, N$ ,

$H_i^t$  represents in time with  $i$ th algal colony,

$N$  represents number of algae.

In AAA, nutrient-rich algal colonies with optimal solutions thrive, and successful traits are transferred from larger colonies to smaller ones through cell replication during the evolutionary process.

Maximum <sup>$t$</sup>  = max  $H_i^t$ , whereas  $i = 1, 2, 3, \dots, N$ ,

Minimum <sup>$t$</sup>  = min  $H_i^t$ , whereas  $i = 1, 2, 3, \dots, N$ ,

Minimum <sup>$t$</sup>  = maximum <sup>$t$</sup> , whereas  $m = 1, 2, 3, \dots, D$ .

In the AAA, algal colonies are ranked by size at time  $t$ . In each dimension, the smallest algal colony's cell dies, while the largest colony's cell replicates itself.

In the AAA algorithm, algal colonies that are unable to grow sufficiently in their environment attempt to adapt by becoming more similar to the largest colony. This process changes the starvation levels within the algorithm. Each artificial alga starts with a starvation value of zero, which increases over time if the algal cell does not receive enough light. The artificial alga with the highest starvation value is the focus of adaptation.

$$Star^t = \max B_i^t, \text{ Where } i = 1, 2, 3, \dots, N \tag{14}$$

$$Star^{t+1} = Star^t + (\max^t - Star^t) * rand \tag{15}$$

Helical movement: The cells and colonies exhibit specific swimming behavior, striving to stay near the water surface where sufficient light for their survival is available. They move in a helical manner, propelled by their flagella, which face limitations from gravity and viscous drag. In the AAA, gravity’s influence is represented by a value of 0, while viscous drag is simulated as shear force, proportional to the size of the algal cell. The cell is modeled as a spherical shape, with its size determined by its volume, and the friction surface is equivalent to the surface area of a hemisphere.

$$\tau(x_i) = 2\pi r^2 \tag{16}$$

$$\tau(x_i) = 2\pi \left( \sqrt[3]{\frac{3H_i}{4\pi}} \right)^2 \tag{17}$$

Friction surface is represented as  $\tau(x_i)$ .

The helical movement of algal cells is determined by three randomly selected dimensions. One dimension corresponds to linear movement, as described by Equation (18). The other two dimensions correspond to angular movement, as described by Equations (19) and (20). Equation (18) is used for one-dimensional problems, allowing the algal cell or colony to move in a single direction. Equation (19) is used for two-dimensional problems, where the algal movement follows a sinusoidal pattern. Equation (20) is used for three or more dimensions, where the algal movement takes on a helical trajectory. The step size of the movement is determined by the friction surface and the distance to the light source.

$$X_{im}^{t+1} = X_{im}^t + (X_{jm}^t - X_{im}^t)(\Delta - \tau^t(X_i))P \tag{18}$$

$$X_{ik}^{t+1} = X_{ik}^t + (X_{jk}^t - X_{ik}^t)(\Delta - \tau^t(X_i))\cos \alpha \tag{19}$$

$$X_{il}^{t+1} = X_{il}^t + (X_{jl}^t - X_{il}^t)(\Delta - \tau^t(X_i))\sin \beta \tag{20}$$

where  $X_{im}^{t+1}, X_{ik}^{t+1}, X_{il}^{t+1}$  represents the x, y, and z coordinates of the *i*th algal cell at time *t*.

The variables  $\alpha$  and  $\beta$  are in the range  $[0, 2\pi]$ , while *p* is within the interval  $[-1, 1]$ .  $\Delta$  represents the shear force and  $\tau^t(X_i)$  denotes the surface area of the *i*th algal cell.

### 5.2. Statistical Analysis

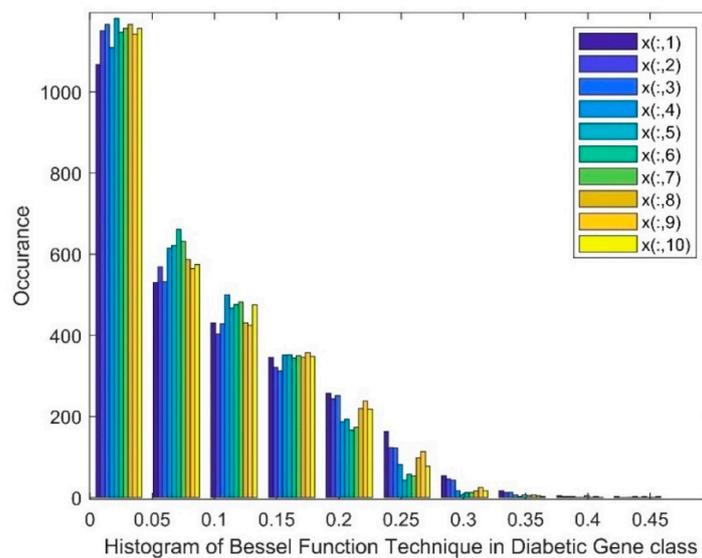
The microarray gene data were reduced in dimension through four distinct dimensionality reduction (DR) techniques and comprehensive analysis using the statistical metrics of mean, variance, skewness, kurtosis, PCC, and CCA. This scrutiny aimed to ascertain whether the outcomes accurately portrayed the inherent properties of microarray genes within the reduced subspace. As shown in Table 2, the DR method based on AAA exhibited elevated mean and variance values across classes. In contrast, the remaining three DR methods—namely the Bessel function, Discrete Cosine Transform (DCT), and Least Squares Linear Regression (LSLR)—revealed modest and overlapping mean and variance values

within classes. Among these methods, the LSLR DR approach showcased negative skewness, indicating the occurrence of skewed elements in the classes. Additionally, the DCT and LSLR DR methods demonstrated negative kurtosis, signifying their preservation of the underlying microarray gene traits. The PCC values revealed substantial correlations within the obtained outputs for a particular class. In the case of the Bessel function DR method, all four statistical parameters exhibited positive values at their minimum. This indicates an association with non-Gaussian and nonlinear distributions, a conclusion substantiated by the histograms, normal probability plots, and scatter plots of the DR method outputs. Canonical Correlation Analysis (CCA) provided insight into the correlation between DR outputs for diabetic and non-diabetic instances. Notably, the low CCA value in Table 2 suggests a limited correlation between the DR outputs of the two distinct classes.

**Table 2.** Statistical analysis for different DR techniques.

Statistical Parameters	Bessel Function		Discrete Cosine Transform (DCT)		Least Squares Linear Regression (LSLR)		Artificial Algae Algorithm (AAA)	
	Dia	Norm	Dia	Norm	Dia	Norm	Dia	Norm
Mean	0.082961	0.084162	1.882012	1.883618	0.00467	0.00457	121.664	120.5492
Variance	0.005165	0.005378	0.50819	0.506957	0.000432	0.000417	101.6366	103.0168
Skewness	0.865169	0.856162	0.187903	0.228924	0.003787	−0.0315	0.042744	0.054472
Kurtosis	0.180926	0.135504	−0.34524	−0.40687	−0.16576	−0.08667	0.152272	0.091169
Pearson CC	0.866264	0.859211	0.98138	0.983118	0.975446	0.977318	0.9826	0.985246
CCA	0.05904		0.260275		0.090825		0.082321	

Figure 3 shows a histogram of the Bessel function DR techniques in the diabetic class. The histogram depicts that a skewed group of values, a gap, and the existence of nonlinearity were witnessed in this method. Patients from 1 to 10 are represented as  $x(:,1)$  to  $x(:,10)$ .



**Figure 3.** Histogram of Bessel function technique in the diabetic gene class.

Figure 4 exhibits a histogram of the BF DR techniques in the non-diabetic class, in which the marker of  $x(:,1)$  represents patient 1 and  $x(:,10)$  represents patient 10. Figure 4 shows a skewed group of values, a gap, and the existence of nonlinearity in this method.

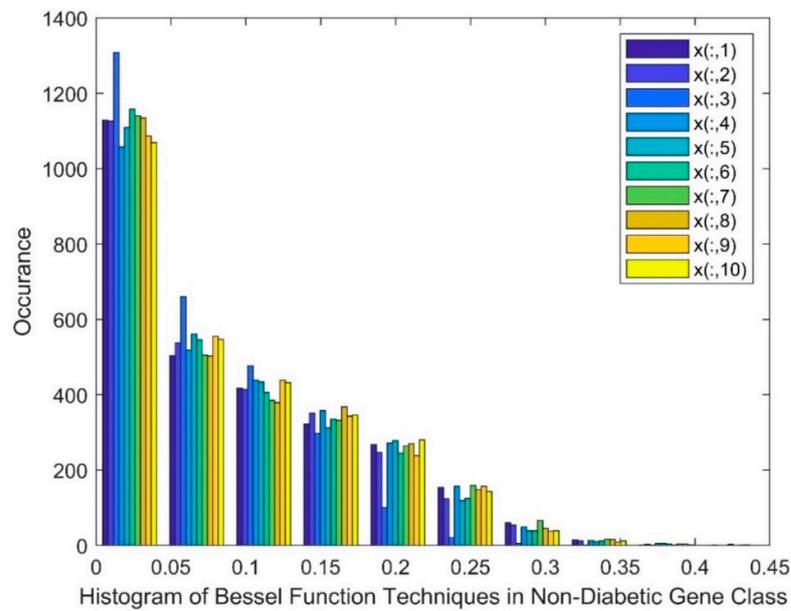


Figure 4. Histogram of Bessel function technique in the non-diabetic gene class.

In Figure 5, data points 1 to 5 signify the reference points, 6 to 10 highlight the upper bound, and 11 to 15 depict the clustered variable points. This representation signifies the generation of a normal probability plot for features obtained using DCT DR techniques within the diabetic gene class. As can be observed from Figure 5, the plot effectively showcases the complete cluster of DCT DR outputs, accentuating the existence of variables with the nature of nonlinearity across classes.

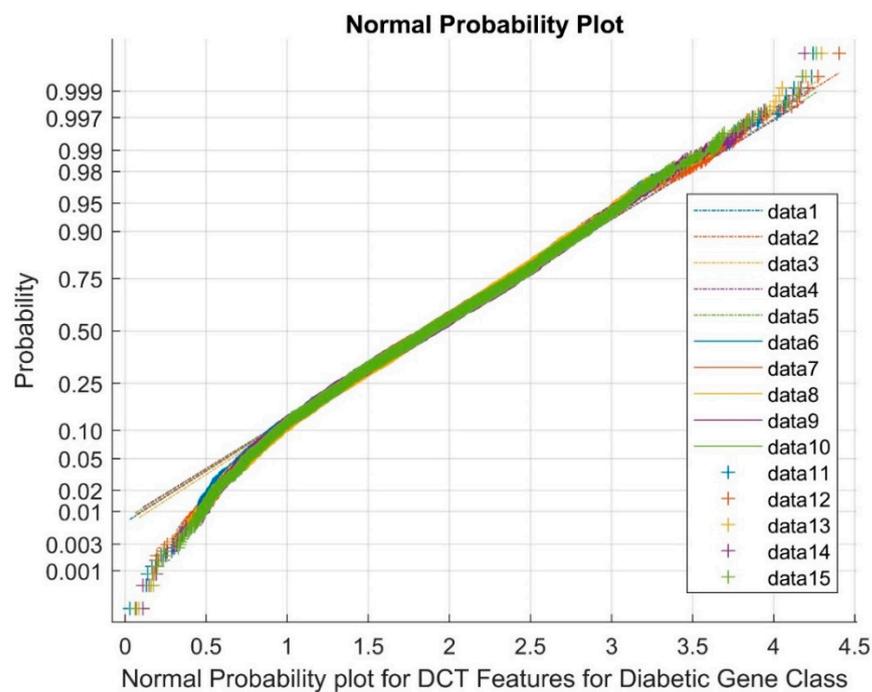
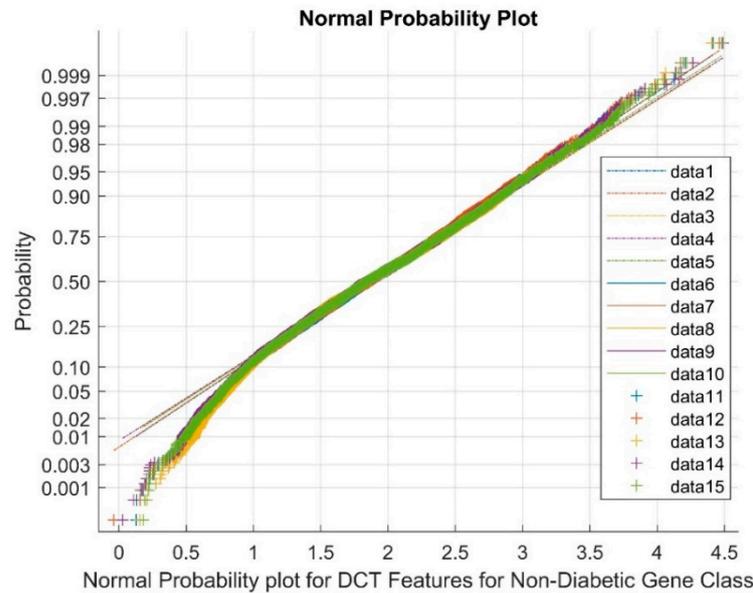


Figure 5. Normal probability plot showcasing DCT features for the diabetic gene class.

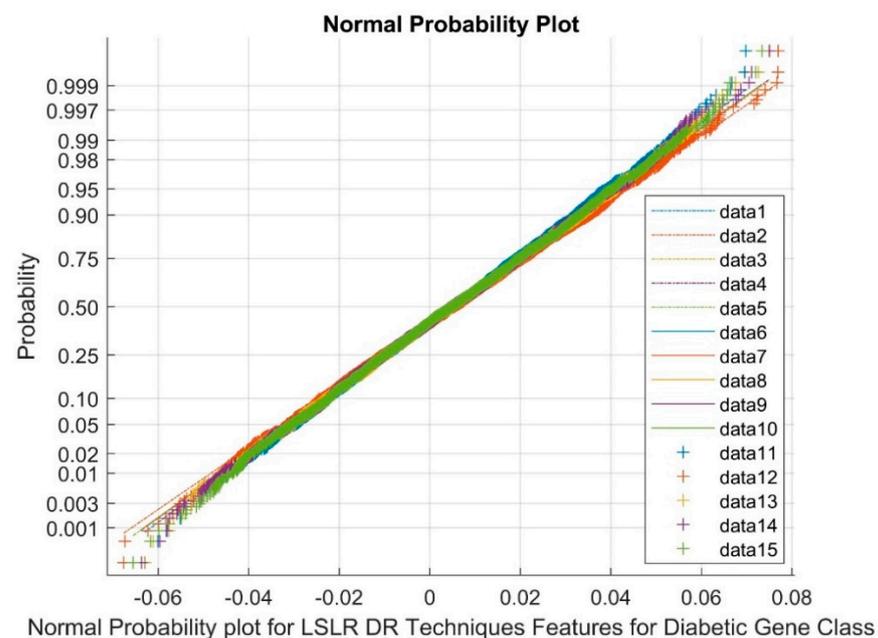
Figure 6 shows the normal probability plot for the DCT DR techniques for the non-diabetic gene class. The data points from 1 to 5 represent references, the upper bound values are represented from 6 to 10, and the cluster variable points are from 11 to 15. The plot shows that the total cluster of DCT DR outputs and nonlinearly correlated variables

among the classes were observed due to the low values of mean and variance and the presence of negative kurtosis variables in the DR method.



**Figure 6.** Normal probability plot representing DCT features for the non-diabetic gene class.

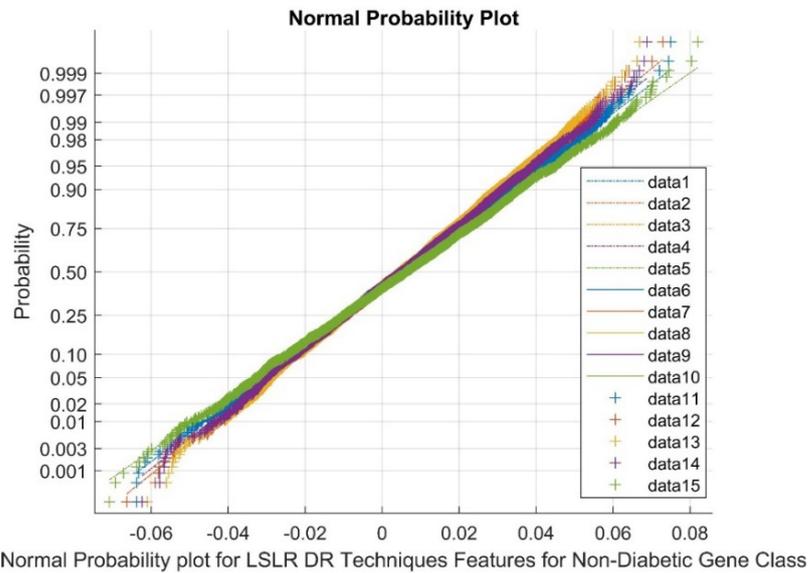
Figure 7 presents data points 1 to 5 as references, 6 to 10 as upper bound values, and 11 to 15 as variable points. The normal probability plot distinctly exhibits clustered groups corresponding to LSLR DR outputs. This observation underscores the existence of non-Gaussian and nonlinearly varying variables among the classes. This phenomenon can be attributed to the low variance and negative kurtosis attributes of the outcomes generated by the DR method.



**Figure 7.** Normal probability plot for LSLR DR techniques in diabetic gene class.

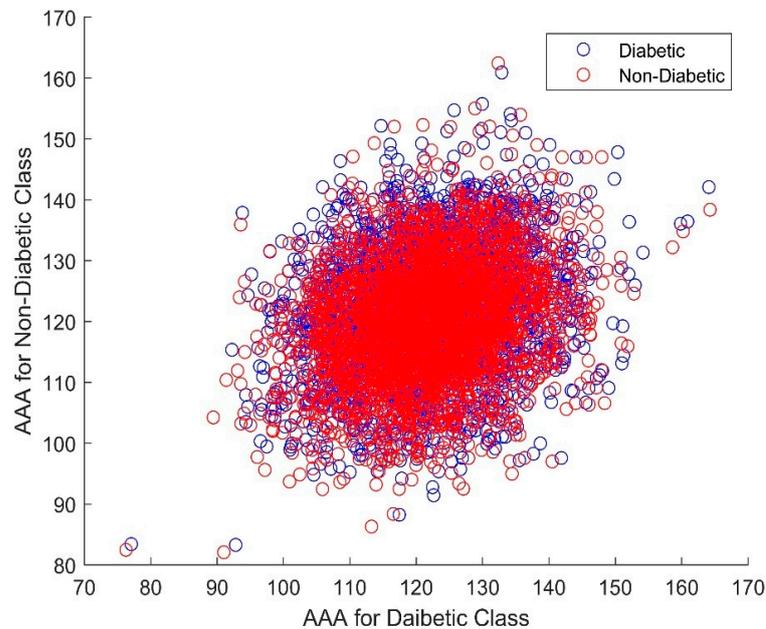
Figure 8 presents the normal probability plot for LSLR DR techniques in the non-diabetic class. The plot displays a discrete group of clusters for LSLR DR outputs. The data points 1 to 5 represent references, 6 to 10 represent upper bound values, and 11 to

15 represent variable points. The flat kurtosis variable and low variance in the DR methods indicate the presence of nonlinearity and a non-Gaussian nature.



**Figure 8.** Normal probability plot for LSLR DR techniques in non-diabetic gene class.

Figure 9 presents a scatter plot of the AAA DR techniques for the non-diabetic and diabetic gene classes. As can be seen, there is total clustering and overlapping of the variables in both classes. The non-Gaussian and nonlinear nature can also be observed from this graph. Furthermore, the AAA algorithm has a heavy computational cost on the classifier design. To reduce the burden of the classifiers, a feature selection process comprising the Elephant Herd Optimization (EHO) and Dragonfly algorithms was initiated.



**Figure 9.** Scatter plot depicting AAA DR results for both non-diabetic and diabetic classes.

### 6. Feature Selection Methods

The reduced dimensionality dataset was used for the feature selection methods. The metaheuristic algorithms of Monarch Butterfly Optimization (MBO) [28], Slime Mold Algorithm (SMA) [29], Moth Search Algorithm (MSA) [30], Hunger Games Search (HGS) [31],

Runge Kutta Method (RUN) [32], Colony Predation Algorithm (CPA) [33], weighted-meanOfvectors (INFO) [34], Harris Hawks Optimization (HHO) [35], Rime Optimization Algorithm (RIME) [36], Elephant Herding Optimization (EHO) [37] algorithm, and Dragonfly Optimization Algorithm (DOA) [38] were considered for the FS.

MBO has two operators: migration and butterfly adjusting operator. The Lévy flight is used in the butterfly adjusting operator, which has infinite mean and variance. SMA is used for attaining global optimization. It has three stages: the first is to make a better solution approach based on the slime mold bound condition through the iterations attained from the tanh function; the second is wrap food, based on SMA, that imitates the updating position of the slime mold; and the third is an oscillator, based on step size, which is considered within bound. MSA was also used to find the global optimization. Moths have the propensity to follow Lévy flights. It exhibits similar characteristics to MBO such as being non-Gaussian and having infinite mean and infinite variance. HGS is a good population-based optimizer; however, when dealing with challenging optimization problems, the classic HGS sometimes shows premature convergence and stagnation shortcomings. Therefore, finding approaches that enhance solution diversity and exploitation capabilities is crucial. RUN is also an optimization technique. Although RUN has a solid mathematical theoretical foundation, there are still some performance defects when dealing with complex optimization problems. In the initialization phase, the focus is on constructing a population that evolves over several iterations. CPA has taken inspiration from the predatory habits of groups in nature. However, CPA suffers from poor exploratory ability and cannot always escape certain solutions. Two strategies are used in the pursuit process to increase the probability of successful predation: scattering prey and surrounding prey. Prey dispersal drives the prey in different directions and weakens the prey group. The weighted-meanOfvectors (INFO) algorithm is also a population-based optimization algorithm operating based on the calculation of the weighted mean for a set of vectors. It has three techniques to update the vectors' location: a local search, a vector-combining rule, and the weighted mean concept for a solid structure. The INFO algorithm's reliance on weighted mean vectors may not capture nonlinear relationships between features and target variables effectively. It focuses on selecting individual features based on their weighted mean values, so may not effectively explore interactions or combinations of features. HHO is a computational intelligence tool, and its complexity may increase with the number of features in high-dimensional datasets. It may struggle to handle large feature spaces efficiently, leading to longer execution times. It replicates Harris hawk predator-prey dynamics. It is divided into three sections: exploring, transformation, and exploitation. It has a high convergence rate and a powerful global search capability, but it has an unsatisfactory optimization effect on high-dimensional or complex problems. RIME is also a good optimization algorithm for search space mechanisms and the typical idea is to compare the updated fitness value of an agent with the global optimum; if the updated value is better than the current global optimum, then the optimum fitness value is replaced, and the agent is recorded as the optimum. The advantage of such an operation is that it is simple and fast, but it does not help in the exploration and exploitation of the population and only serves as a record. However, algorithms like EHO and DOA are used as feature selection parameters for emulating the behavior observed in elephants and dragonflies for the better selection of features and offer effective approaches to address the abovementioned challenges in optimization techniques for FS.

#### Elephant Herding Optimization (EHO) algorithm

Wang et al. [37] introduced EHO as a metaheuristic algorithm inspired by the behavior of elephants in the African savanna. It has demonstrated effectiveness in solving optimization problems and has been successfully applied in various domains, including feature selection. In feature selection, the objective is to identify a subset of informative features from a larger set that are relevant to the target variable. EHO employs a herd of elephants to search for the optimal solution, with each elephant representing a potential solution. By combining global and local search strategies, the algorithm guides the elephants towards

the best solution. The methodology of the EHO is depicted in Figure 10. EHO offers immense potential as a feature selection technique due to its ability to strike a balance between global and local searches, making it suitable for high-dimensional data. The initialization of the elephant herd involves assigning random positions to the elephants in the feature space, providing a comprehensive representation of the elephants' positions and the overall movement of the herd.

$$y_i^{new} = y_i^{old} + \alpha (Y_{best} - Y_i^{old}) * r \tag{21}$$

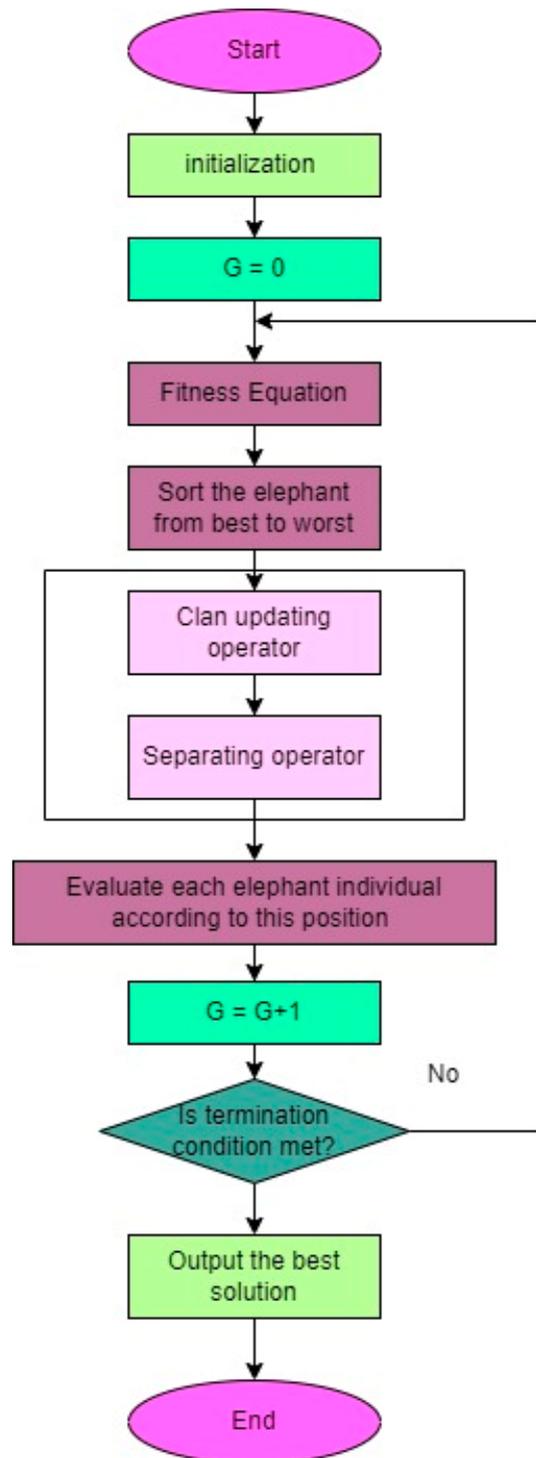


Figure 10. Diagram illustrating the process of the EHO algorithm.

The EHO algorithm [39] involves updating the positions of elephants within the herd. This update process considers both the old position ( $y_i^{old}$ ) and the new position ( $y_i^{new}$ ) of each elephant. A control parameter ( $\alpha$ ), which falls within the range of [0, 1], is used in conjunction with a randomly generated number ( $r \in [0, 1]$ ) to determine the new position. Additionally, each elephant in the herd maintains a memory of its best position in the feature space. The best position is updated using the following equation, ensuring that the elephant’s memory is updated accordingly.

$$Y_{best} = \beta * Y_{centre} \tag{22}$$

$$Y_{centre} = \frac{1}{m} * \sum_{i=1}^m y_i \tag{23}$$

The algorithm includes the concept of the best position ( $Y_{best}$ ) for each elephant within the herd. This best position is determined by considering the control parameter ( $\beta$ ), which falls within the range of [0, 1]. The control parameter plays a role in updating and adjusting the best position of the elephant, ensuring that it reflects the optimal solution obtained during the optimization process.

By considering both the best and worst solutions, the EHO algorithm ensures a more comprehensive exploration of the solution space, leading to improved optimization performance.

$$Y_{worst} = Y_{min} + (Y_{max} - Y_{min} + 1) * rand \tag{24}$$

#### Dragonfly Optimization Algorithm (DOA)

The Dragonfly Algorithm (DA) is an optimization technique based on swarm intelligence, taking inspiration from the collective behaviors of dragonflies. Introduced by Mirjalili in 2016 [38], this algorithm mimics both static and dynamic swarming behaviors observed in nature. Figure 11 shows the flowchart of DOA. During the dynamic or exploitation phase, it forms large swarms and travels in a specific direction to confuse potential threats. In the static or exploration phase, the swarms form smaller groups, moving within a limited area to hunt and attract prey [40]. The DA is guided by five fundamental principles: separation, alignment, cohesiveness, attraction, and diversion. These principles dictate the behavior of individual dragonflies and their interactions within the swarm. In the equations that follow,  $K$  and  $K_i$  denote the current position and the  $i$ th position of a dragonfly, respectively, while  $N$  represents the total number of neighboring flies.

**Separation:** This implies that the static phase of the algorithm focuses on preventing dragonflies from colliding with each other in their vicinity. This calculation aims to ensure the avoidance of collisions among flies.

$$Se_j = -\sum_{i=1}^n k - k_j \tag{25}$$

where  $Se_j$  represents the motion of the  $i$ th individual aimed at maintaining separation from other dragonflies.

**Alignment:** This denotes the synchronization of velocities among dragonflies belonging to the same group. It is represented as

$$Ag_j = \frac{\sum_{i=1}^n Ve_i}{n} \tag{26}$$

This is represented by  $Ag_j$ , which is called the velocity of the  $i$ th individual.

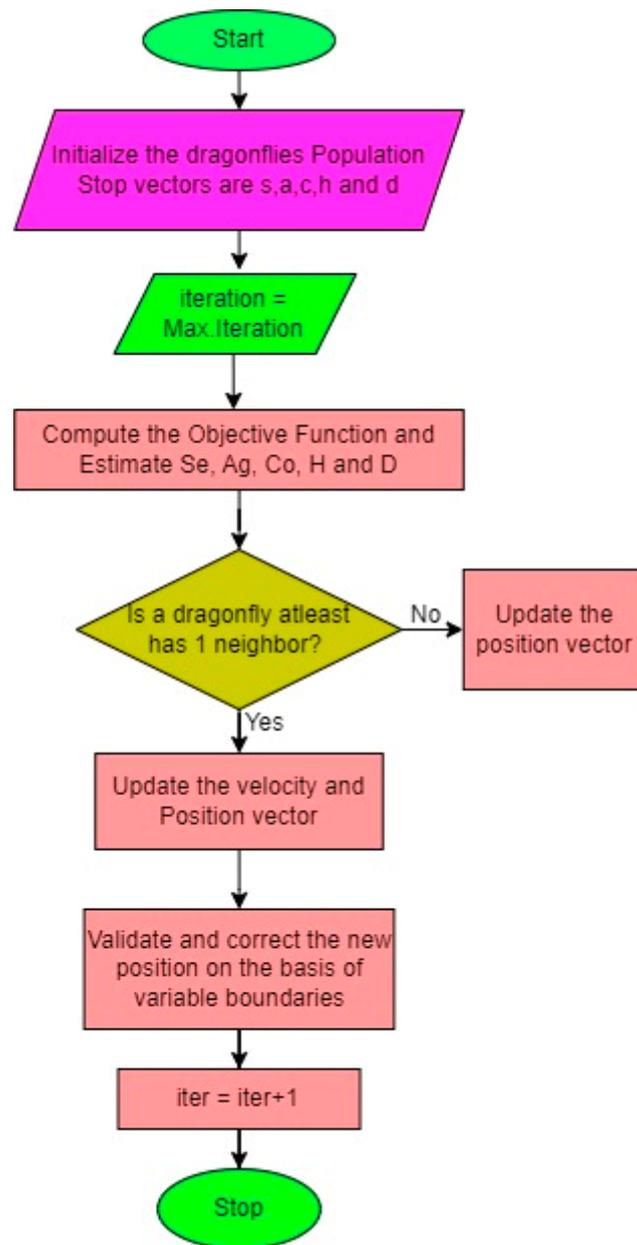


Figure 11. Flowchart of the Dragonfly Optimization algorithm.

Cohesiveness: This represents the inclination of individual flies to converge towards the center of swarms. The calculation is

$$Co_j = \frac{\sum_{i=1}^n k_i}{N} - k \tag{27}$$

Attraction: The quantification of the attraction towards the food source is characterized by

$$H_j = K^+ - K \tag{28}$$

Here,  $H_j$  is the attraction of the food source, and  $K^+$  represents the position of the food source.

Diversion: The diversion from the enemy is determined by the outward distance, which is calculated as

$$D_j = K^- + K \tag{29}$$

The calculation of the outward distance determines the diversion from the enemy, and it is expressed in terms of the step vector ( $\Delta K$ ) and the current position vector ( $K$ ) are used to update the locations of artificial dragonflies within the search space. The step vector ( $\Delta K$ ) can be calculated using the direction of movement of the dragonfly:

$$\Delta K_j^{t+1} = (sSe_j + aAg_j + cCo_j + hH_j + dD_j) + \omega\Delta K_j^t \tag{30}$$

The behavior of the dragonfly algorithm is influenced by factors such as separation weight ( $s$ ), alignment weight ( $a$ ), cohesion weight ( $c$ ), attraction weight ( $h$ ), and enemy weight ( $d$ ). The inertia weight is represented by " $\omega$ ", and " $t$ " represents the iteration number.

Through the manipulation of these weights, the algorithm can attain both exploration and exploitation phases. The position of the  $i$ th dragonfly at  $t + 1$  iterations is determined by the following equation:

$$K_j^{t+1} = K_j^t + \Delta K_j^{t+1} \tag{31}$$

The evaluation of this method's outcomes is conducted by assessing the consequence of the  $p$ -value using the  $t$ -test. Table 3 demonstrates the significance of the  $p$ -values associated with the EHO and Dragonfly Algorithm methods across the four DR techniques. The data presented in Table 3 reveal that both the EHO and Dragonfly Algorithms' feature selection methods do not exhibit significant  $p$ -values across classes for all four dimensionality reduction methods. This  $p$ -value serves as an initial indicator to quantify the existence of outliers, nonlinearity, and non-Gaussian nature among the classes after the implementation of feature selection techniques.

**Table 3.** Significance of  $p$ -values for feature selection methods using  $t$ -test across various DR techniques.

Feature Selection	DR Techniques	Bessel Function		Discrete Cosine Transform (DCT)		Least Squares Linear Regression (LSLR)		Artificial Algae Algorithm (AAA)	
	Genes	Dia	Norm	Dia	Norm	Dia	Norm	Dia	Norm
EHO	$p$ -value < 0.05	0.9721	0.9998	0.994	0.9996	0.9961	0.9999	0.9466	0.9605
Dragonfly	$p$ -value < 0.05	0.99985	0.876	0.9956	0.998	0.9951	0.99931	0.9936	0.9977

### 7. Classification Techniques

#### NLR—Nonlinear Regression

The behavior of a system is expressed through mathematical equations to facilitate representation and analysis, ultimately aiming to determine an exact best-fit line between classifier values. Nonlinear regression introduces nonlinear and random variables ( $a, b$ ) to capture the complexity of the system. The primary objective of nonlinear regression is to reduce the sum of squares. This involves measuring values from the dataset and computing the difference between the mean and each data point, squaring these differences, and summing them. The minimum value of the sum of squared differences indicates a better fit to the dataset.

Nonlinear models require more attention due to their inherent complexity, and researchers have devised various methods to mitigate this difficulty, such as the Levenberg–Marquardt and Gauss–Newton methods. Estimating parameters for nonlinear systems is achieved through least squares methods, aiming to minimize the residual sum of squares. Iterative techniques, including the Taylor series, steepest descent method, and Levenberg–Marquardt method (Zhang et al. [41]), can be employed for nonlinear equations. The Levenberg–Marquardt technique is commonly used for assessing the nonlinear least squares, offering advantages and producing reliable results through an iterative process.

The authors assume a represented model:

$$z_i = f(x_i, \theta) + \varepsilon_i, \text{ where } i = 1, 2, 3, \dots, n \tag{32}$$

Here,  $x_i$  and  $z_i$  represent the individual and supported variables of the  $i$ th iteration,  $\theta = (\theta_1, \theta_2, \dots, \theta_m)$  are the parameters, and  $\varepsilon_i$  are the error terms that follow  $N(0, \sigma^2)$ .

$$S_u(\theta) = \sum_{i=1}^n [z_i - f(x_i, \theta)]^2 \tag{33}$$

Let  $\theta_k = \theta_{1k}, \theta_{2k}, \dots, \theta_{pk}$  be the starting values; the successive estimates are obtained using

$$(H + \tau I)(\theta_0 - \theta_1) = g \tag{34}$$

where  $g = \frac{\partial S_u(\theta)}{\partial \theta} \Big|_{\theta = \theta_0}$  and  $H = \frac{\partial^2 S_u(\theta)}{\partial \theta \partial \theta'} \Big|_{\theta = \theta_1}$ ,  $\tau$  are a multiplier and  $I$  is the identity matrix.

The integrity of the model is assessed using the MSE, which quantifies the discrepancy between the experimental and estimated values. The MSE is computed as the average squared difference between the actual and predicted values. The overall experimental values are expressed in terms of  $N$ .

$$MSE = \frac{1}{N} \sum_{(i=1)}^N (y_i - y_i^\Lambda)^2 \tag{35}$$

The steps for nonlinear regression are the initialization of the initial parameters and the generation of curves based on these values. The goal is to iteratively modify the parameters to minimize the MSE and bring the curve closer to the desired value. The process continues until the MSE value no longer changes compared to the previous iteration, indicating convergence.

#### Linear Regression (LR)

In the investigation of gene expression data, linear regression is a suitable method for obtaining the best-fit curve, as the conveyance levels in the genes exhibit only minor variations. To identify the most informative genes, a feature selection process is performed by comparing the training dataset with the gene expression data within different levels of diversity. In this linear regression model, the dependent variable, denoted as  $x$ , is associated with the independent variable,  $y$ . The model aims [42] to predict values using the  $x$  variable, optimizing the regression fitness value based on the population in the  $y$  variable. The hypothesis function for a single variable is given by

$$g_\theta = \theta_0 + \theta_1 x \tag{36}$$

where  $\theta_i$  represents the parameters. The objective is to select the range of  $\theta_0$  and  $\theta_1$  that ensures  $g_\theta$  closely approximates  $y$  in the training dataset  $(x, y)$ .

$$R(\theta_0, \theta_1) = \frac{1}{2m} \sum_{i=1}^m (g_\theta(x^i) - y^i)^2 \tag{37}$$

Here, “ $m$ ” symbolizes the total count of samples within the training dataset. For LR models with  $n$  variables, the hypothesis function becomes

$$g_\theta = \theta_0 x_0 + \theta_1 x_1 + \dots + \theta_n x_n \tag{38}$$

and the cost function is given by

$$R(\theta) = \frac{1}{2m} \sum_{i=1}^m (g_\theta(x^i) - y^i)^2 \tag{39}$$

where  $\theta$  is a set of parameters  $\{\theta_0, \theta_1, \theta_2, \dots, \theta_n\}$ . The gradient descent algorithm is employed to minimize the cost function, and the partial derivative of the cost function is computed as

$$\frac{\delta}{\delta\theta_j} g(\theta) = \frac{\delta}{\delta\theta_j} \sum_{i=1}^m (g_\theta(x^i) - y^i)^2 \tag{40}$$

To update the parameter value  $\theta_j$ , the following equation is used:

$$\theta_{j(new)} = \theta_{j(old)} + \beta \frac{1}{m} \sum_{i=1}^m (g_\theta(x^i - y^i) x_j^i) \tag{41}$$

where  $\beta$  represents the learning rate, and  $\theta_j$  is continuously computed until convergence is reached. In this study,  $\beta$  is set to 0.01.

The algorithm for LR involves the following steps:

Feature selection parameters, obtained from algorithms such as the Bessel function, DCT, LSLR, and AAA, are used as input for the classifiers.

A line represented by  $g_\theta = \theta_0 + \theta_1 x$  is fitted to the data in a linear manner.

The cost function is formulated with the aim of minimizing the squared error existing between the observed data and the predictions.

The solutions are found by equating the derivatives of  $\theta_0$  and  $\theta_1$  to zero.

To yield the coefficient of MSE, repeat steps 2, 3, and 4.

#### Gaussian Mixture Model (GMM)

GMM is a well-known unsupervised learning technique in machine learning used for many applications like pattern recognition and signal classifications. It involves integrating related objects based on clustering techniques. By classifying the data, GMM [43] facilitates the prediction and computation of unrated items within the same category. Hard and soft clustering techniques are used by GMM, and it utilizes the distribution for data analysis. Each GMM consists of multiple Gaussian distributions (referred to as "g"). The PDF of GMM combines these distributed components linearly, enabling easier analysis of the generated data. When generating random values as a vector "a" within an n-dimensional sample space  $\chi$ , if "a" adheres to a Gaussian distribution, the expression for its probability distribution function is as follows:

$$p(a) = \frac{1}{(2\pi)^{n/2} |\Sigma|^{1/2}} e^{-\frac{1}{2}(a-\mu)^T \Sigma^{-1} (a-\mu)} \tag{42}$$

Here,  $\mu$  represents the mean vector in the n-dimensional space, and  $\Sigma$  is the covariance matrix of size  $n \times n$ . The determination of the covariance matrix and mean vector is essential for the Gaussian distribution. Multiple components are mixed in the Gaussian distribution function [44], with each component and the equation of mixture distribution given by

$$P_Q(a) = \sum_{i=1}^k \alpha_j \times p(a|\mu_j, \Sigma_j) \tag{43}$$

In this equation,  $\alpha_j$  represents the mixing coefficient corresponding to the  $j$ th Gaussian mixture, while  $\mu_j$  and  $\Sigma_j$  denote the mean vector and covariance matrix of that Gaussian component, respectively.

#### Expectation Maximum (EM)

The EM algorithm [45] serves as a classifier in this context. Its primary objective is to estimate missing values within a dataset and subsequently predict those values to maximize the dataset's order based on the application's requirements. Consider two random variables,  $X$  and  $Y$ , involved in the prediction process and determining the order

of the data in rows. Variable  $X$  is observable and known in the dataset, while the unknown variable  $Z$  needs to be predicted to set the value of  $Y$ .

$$L(\theta; X, Y) = p(X, Y|\theta) \tag{44}$$

$$L(\theta|x) \in \{ \alpha p(X|\theta); \alpha > 0 \}. p(X|\theta) * p(Y|\theta) \tag{45}$$

The maximum likelihood estimation is obtained as

$$L(\theta; X) = p(X|\theta) = \sum_Y p(X, Z|\theta) \tag{46}$$

To estimate the expected value of the log-likelihood function, we calculate

$$Q(\theta|\theta^{\wedge}(t)) = E_{z|x; \theta^{\wedge}(t)} [\log L(\theta; X, Y)] \tag{47}$$

The above quantity is maximized to compute the maximum value, resulting in

$$\theta^{\wedge}(t + 1) = \arg \max Q(\theta|\theta^{\wedge}(t)) \tag{48}$$

The expectation and maximization steps are iteratively repeated until a converged sequence of values is reached as mentioned in Figure 12 which is the flow diagram of expectation maximization.

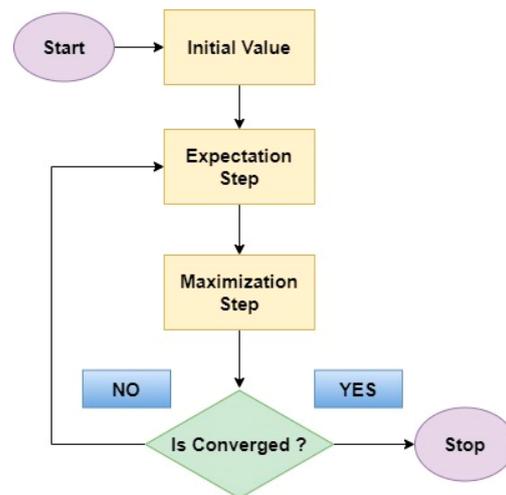


Figure 12. Flow diagram of Expectation Maximum.

### Bayesian Linear Discriminant Classifier (BLDC)

BLDC [46] is commonly employed to regularize high-dimensional signals, reduce noisy signals, and improve computational efficiency. Before conducting Bayesian linear discriminant analysis, an assumption is made that a target, denoted as  $b$ , is related to a vector  $x$  with the addition of white Gaussian noise,  $c$ .

This relationship can be expressed as  $a = x^T b + c$ . The function  $x$  is assigned weights, and its likelihood function is given by

$$p(G|\beta, x) = \left(\frac{\beta}{2\pi}\right)^{\frac{c}{2}} \exp\left(-\frac{\beta}{2}\|B^T x - m\|\right) \tag{49}$$

where the pair  $\{B, m\}$  represents  $G$ . The  $x$  of prior distribution is expressed as

$$p(x|\alpha) = \left(\frac{\alpha}{2\pi}\right)^{\frac{1}{2}} \left(\frac{\varepsilon}{2\pi}\right)^{\frac{1}{2}} \exp\left(-\frac{1}{2}x^T H'(\alpha)x\right) \tag{50}$$

The regularization square matrix is given by

$$H'(\alpha) = \begin{bmatrix} \alpha & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \varepsilon \end{bmatrix}_{(l+1)(l+1)} \tag{51}$$

and  $\alpha$  is a hyperparameter obtained from data forecasting, while  $l$  represents the assigned vector number. By applying Bayes' rule,  $x$  can be calculated as

$$p(x|\beta, \alpha, G) = \frac{P(G|\beta, x)P(x|\alpha)}{\int P(G|\beta, x)P(y|\alpha)dy} \tag{52}$$

The mean vector  $v$  and the covariance matrix  $X$  must adhere to the specific norms outlined in Equations (50) and (52) for the posterior distribution. The predominant nature of the posterior distribution is Gaussian.

$$v = \beta(\beta BB^T + H'(\alpha))^{-1} Ba \tag{53}$$

$$X = (\beta BB^T + H'(\alpha))^{-1} \tag{54}$$

When predicting the input vector  $\hat{b}$ , the probability distribution for regression can be expressed as

$$p(\hat{a}|\beta, \alpha, \hat{b}, \widehat{G}) = \int p(\hat{a}|\beta, \hat{b}, x)p(x|\beta, \alpha, G)dy \tag{55}$$

Again, the nature of this prediction analysis is predominantly Gaussian, with the mean expressed as  $\mu = v^T \hat{b}$  and variance expressed as  $\delta^2 = \frac{1}{\beta} + \hat{b}^T X \hat{b}$ .

### Logistic Regression (LoR)

Logistic Regression (LoR) has proven to be effective in classifying diseases such as diabetes, types of cancer, and epilepsy. In this context, function  $y$  that represents the disease level is considered, ranging from 0 to 1 to indicate non-diabetic and diabetic patients, respectively. Gene expressions are represented by a vector  $x = x_1, x_2, \dots, x_m$ , where each element  $x_j$  corresponds to the expression level of the  $j$ th gene. Using a model-based approach for a dataset  $\Pi(x)$ , the aim is to identify informative genes for diabetic patients based on the likelihood of  $y$  being 1 given  $x$ . To achieve dimensionality reduction, logistic regression is utilized to select the most relevant “ $q$ ” genes. The gene expression representation  $x_j^*$  corresponds to the gene expression, with  $j$  ranging from 1 to  $q$ , while the binary disease status is denoted by  $y_i$ , where  $i$  ranges from 1 to  $n$ . The logistic regression model can be expressed as

$$Logit\{\Pi(x)\} = v_0 + \sum_{j=1}^q v_j x_j^* \tag{56}$$

The objective is to maximize the fitness and log-likelihood, which can be achieved by obtaining the following function

$$1(v_0, v) = \sum_{j=1}^n \{y_i \log(\pi_i) + (1 - \pi_i)\} - \frac{1}{2\tau^2} \|v\|^2 \tag{57}$$

where  $\tau$  is a parameter that limits the reduction in  $v$  near 0,  $\pi_i = \pi(x_i)$  as defined by the model [47,48], and  $\|v\|^2$  denotes the Euclidean length of  $v = v_1, v_2, \dots, v_p$ . The selection of  $q$  and  $\tau$  are determined using the parametric bootstrap method, which imposes constraints on accurate error prediction. Initially,  $v = 0$ , for the purpose of calculating the cost function. It is then varied with different parameters to minimize the cost function. The sigmoid function is applied to restrict values between 0 and 1, serving as an attenuation mechanism.

The threshold cut-off value of 0.5 is used to classify patients as either diabetic or non-diabetic. Any probability below the threshold is considered indicative of non-diabetic patients, while values above the threshold indicate diabetic patients.

SDC—Softmax Discriminant Classifier

The SDC is used to verify and detect the group to which a particular test sample belongs [49]. It weighs the distance between the training samples and the test sample within a particular class or group of data.  $Z$  is represented as

$$Z = [Z_1, Z_2, \dots, Z_q] \in \mathfrak{R}^{c \times d} \tag{58}$$

consisting of samples from distinct classes named  $q$ ,  $Z_q = [Z_1^q, Z_2^q, \dots, Z_{d_q}^q] \in \mathfrak{R}^{c \times d_q}$ . Each class, represented by  $Z_q$ , contains samples from the  $q$ th class, where  $\sum_{i=1}^q d_i = d$ . The sum of the sample sizes, given a test sample  $K \in \mathfrak{R}^{c \times 1}$ , is passed through the classifiers to obtain minimal construction errors, thereby assigning it to the class  $q$ . The transformation of class samples and test samples in SDC involves nonlinear enhancement values. This is achieved through the following equations:

$$h(K) = \operatorname{argmax}_w Z_w^i \tag{59}$$

$$h(K) = \operatorname{argmax}_i \log \left( \sum_{j=1}^{d_i} \exp \left( -\lambda \|v - v_j^i\|_2 \right) \right) \tag{60}$$

In these equations,  $h(K)$  represents the differentiate of the  $i$ th class and  $\|v - v_j^i\|_2$  approaches zero, resulting in the maximization of  $Z_w^i$ . This asymptotic behavior leads to the maximum likelihood of the test sample belonging to a particular class.

Support Vector Machines

The SVM classifier is a significant machine learning approach widely used for classification problems, particularly in the phase of nonlinear regression [50]. In this study, three distinct methods are explored for data classification:

SVM-Linear: this method utilizes a linear kernel to classify the data.

SVM-Polynomial: this approach involves the use of a polynomial kernel for data classification.

SVM-Radial Basis Function (RBF): the RBF kernel is used here to classify the data.

These three SVM methods offer different strategies for effectively classifying datasets, allowing researchers to choose the most suitable approach based on their specific classification requirements.

The training time and computational complexity of the SVM depend on the data and classifiers used. When the number of supports in the SVM increases, it results in higher computational requirements due to the calculation of floating-point multiplications and additions. To address this issue, K-means clustering techniques have been introduced to reduce the number of supports in the SVM. In the linear case, Lagrange multipliers can be employed, and the data points on the borders are expressed as  $v = \sum_{i=1}^m \alpha_i z_i y_i^T$ . Here,  $m$  represents the number of supports,  $z_i$  represents the target labels for  $y$ , and the linear discriminant function is used.

$$h(y) = \operatorname{sgn} \left( \sum_{i=1}^m \alpha_i z_i y_i^T y + C \right) \tag{61}$$

The process of implementing the Support Vector Machine (SVM) involves several key steps.

*Step 1:* The first step is to use quadratic optimization to linearize and converge the problem. By transforming the primal minimization problem into a dual optimization problem, the objective is to maximize the dual Lagrangian  $L_D$  with respect to  $\alpha_i$ :

$$MaxL_D = \sum_{i=1}^l \alpha_i - \frac{1}{2} \sum_{i=1}^l \sum_{j=1}^l \alpha_i \alpha_j y_i y_j (X_i \cdot X_j) \tag{62}$$

subject to  $\sum_{i=1}^l \alpha_i y_i = 0$ , where  $\alpha_i \geq 0 \forall i = 1, 2, 3, \dots, l$ .

*Step 2:* The next step involves solving the quadratic polynomial programming to obtain the optimal separating hyperplane. The data points with non-zero Lagrangian multipliers ( $\alpha_i > 0$ ) are identified as the support vectors.

*Step 3:* The optimal hyperplane is determined based on the support vectors, which are the data points closest to the decision boundary in the trained data.

*Step 4:* K-means clustering is applied to the dataset, grouping the data into clusters according to the conditions from Steps 2 and 3. Three points are randomly chosen from each cluster as the center points, which are representative points from the dataset. Each center point acquires the points around them.

*Step 5:* When there are six central points, each representing an individual cluster, the SVM training data are acquired through the utilization of kernel methods.

$$\text{Polynomial Function: } K(X, Z) = (X^T Z + 1)^d \tag{63}$$

$$\text{Radial Basis Function : } k(x_i, x_j) = \exp \left\{ \frac{-|x_i - x_j|^2}{(2 * \sigma)^2} \right\} \tag{64}$$

### 7.1. Training and Testing of Classifiers

Due to the limited availability of training data, we employed k-fold cross-validation, a widely used technique for evaluating machine learning models. The methodology described by Fushiki et al. [51] was followed to conduct the k-fold cross-validation. Initially, the dataset was divided into k equally sized subsets or “folds”. This process was repeated for all k-folds, ensuring that each fold was used once for testing. Consequently, k performance estimates (one for each fold) were obtained. To obtain an overall estimate of the model’s performance, the average of these k performance estimates was calculated. After training and validating the model using k-fold cross-validation, it was retrained on the complete dataset to make predictions on new, unseen data. The significant advantage of utilizing this method is the more reliable model performance compared to other test split methods, as the technique maximizes the utilization of the available data. Here, we adopted a k-value of 10-fold cross-validation. Furthermore, the research incorporated 2870 dimensionally reduced features per patient, focusing on a cohort of 20 patients with diabetes and 50 non-diabetic patients. The utilization of cross-validation eliminates any reliance on a specific pattern for the test set, enhancing the robustness of our findings. The training process is regulated in the MSE proposed by Wang et al. [52], which is defined as follows:

$$MSE = \frac{1}{N} \sum_{j=1}^N (O_j - T_j)^2 \tag{65}$$

where  $O_j$  is the observed value at time  $j$ , and  $T_j$  is the target value at model  $j$ .

Table 4 represents confusion matrix for detecting diabetes. The following terms in Table 4 can be defined as:

TP—true positive: a patient is accurately classified into the diabetic class.

TN—true negative: a patient is accurately recognized as belonging to the non-diabetic class.

FP—false positive: a patient is inaccurately classified as belonging to the diabetic class when they actually belong to the non-diabetic class.

FN—false negative: a patient is inaccurately classified as being in the non-diabetic class when they should be categorized as belonging to the diabetic class.

**Table 4.** Confusion matrix for detecting diabetes.

Clinical Situation		Predicted Values	
		Diabetic	Non-Diabetic
Real Values	Diabetic class	TP	FN
	Non-diabetic class	FP	TN

Table 5 provides insight into the performance of the classifiers without the feature selection method, focusing on the training and testing Mean Squared Error (MSE) across various DR techniques. The training MSE values consistently range between  $10^{-4}$  and  $10^{-10}$ , while the testing MSE varies from  $10^{-4}$  to  $10^{-8}$ . Among the classifiers, the SVM (RBF) classifier using the AAA DR technique without feature selection achieves the lowest training and testing MSE, specifically  $1.93 \times 10^{-10}$  and  $1.77 \times 10^{-8}$ , respectively. Notably, a lower testing MSE indicates superior classifier performance. It is evident from Table 5 that higher testing MSE values correspond to lower classifier performance, regardless of the DR techniques used.

**Table 5.** Analysis of MSE in for different DR techniques without feature selection.

Classifiers	Bessel Function		Discrete Cosine Transform (DCT)		Least Squares Linear Regression (LSLR)		Artificial Algae Algorithm (AAA)	
	MSE Training Set	MSE Testing Set	MSE Training Set	MSE Testing Set	MSE Training Set	MSE Testing Set	MSE Training Set	MSE Testing Set
	NLR	$2.3 \times 10^{-6}$	$1.76 \times 10^{-3}$	$6.41 \times 10^{-6}$	$2.48 \times 10^{-5}$	$7.75 \times 10^{-6}$	$5.12 \times 10^{-5}$	$2.91 \times 10^{-7}$
LR	$2.41 \times 10^{-5}$	$9.51 \times 10^{-5}$	$7.52 \times 10^{-6}$	$3.11 \times 10^{-5}$	$2.18 \times 10^{-7}$	$4.66 \times 10^{-5}$	$3.67 \times 10^{-8}$	$1.45 \times 10^{-5}$
GMM	$2.1 \times 10^{-5}$	$1.75 \times 10^{-4}$	$5.72 \times 10^{-7}$	$6.8 \times 10^{-6}$	$3.09 \times 10^{-7}$	$1.11 \times 10^{-5}$	$3.76 \times 10^{-6}$	$5.33 \times 10^{-5}$
EM	<b><math>1.62 \times 10^{-7}</math></b>	<b><math>9.87 \times 10^{-6}</math></b>	$2.71 \times 10^{-6}$	$1.3 \times 10^{-5}$	$9.87 \times 10^{-7}$	$1.99 \times 10^{-5}$	$8.97 \times 10^{-9}$	$7.3 \times 10^{-6}$
BLDC	$1.4 \times 10^{-6}$	$2.53 \times 10^{-3}$	$2.86 \times 10^{-7}$	$3.94 \times 10^{-5}$	$4.74 \times 10^{-6}$	$5.28 \times 10^{-5}$	$1.43 \times 10^{-7}$	$1.64 \times 10^{-5}$
LoR	$1.2 \times 10^{-6}$	$2.89 \times 10^{-3}$	$9.47 \times 10^{-6}$	$3.58 \times 10^{-5}$	$8.69 \times 10^{-6}$	$4.54 \times 10^{-5}$	$9.26 \times 10^{-8}$	$1.45 \times 10^{-5}$
SDC	$1.9 \times 10^{-6}$	$2.03 \times 10^{-3}$	$3.66 \times 10^{-6}$	$1.07 \times 10^{-5}$	$2.47 \times 10^{-6}$	$1.86 \times 10^{-5}$	$2.31 \times 10^{-9}$	$5 \times 10^{-6}$
SVM (L)	$3.1 \times 10^{-6}$	$2.7 \times 10^{-3}$	$8.92 \times 10^{-6}$	$2.89 \times 10^{-5}$	$1.09 \times 10^{-5}$	$4.01 \times 10^{-5}$	$4.13 \times 10^{-9}$	$8.2 \times 10^{-6}$
SVM (Poly)	$3.6 \times 10^{-5}$	$2.11 \times 10^{-3}$	$3.36 \times 10^{-6}$	$2.11 \times 10^{-5}$	$1.29 \times 10^{-6}$	$2.85 \times 10^{-5}$	$7.84 \times 10^{-9}$	$4.69 \times 10^{-6}$
SVM (RBF)	<b><math>4.16 \times 10^{-7}</math></b>	<b><math>8.3 \times 10^{-5}</math></b>	<b><math>1.57 \times 10^{-8}</math></b>	<b><math>2.41 \times 10^{-6}</math></b>	<b><math>3.22 \times 10^{-8}</math></b>	<b><math>5.64 \times 10^{-6}</math></b>	<b><math>1.93 \times 10^{-10}</math></b>	<b><math>1.77 \times 10^{-8}</math></b>

Table 6 exhibits the training and testing of MSE in the classifiers with EHO feature selection method across all four DR techniques. The training MSE varies from  $10^{-5}$  to  $10^{-10}$ , while the testing MSE varies between  $10^{-5}$  and  $10^{-8}$ . The SVM (RBF) classifier in the AAA DR method with PSO feature selection achieved a minimum training and testing MSE of  $1.99 \times 10^{-10}$  and  $2.5 \times 10^{-8}$ , respectively. The Bessel function DR method indicates slightly lower training and testing MSE values for the classifiers when compared to the other three DR techniques. All of the classifiers had slightly enhanced testing performance when compared to methods without feature selection. This indicates the enhancement in classifier performance irrespective of the DR technique.

**Table 6.** Analysis of MSE performance for classifiers using EHO feature selection methods across different DR techniques.

Classifiers	Bessel Function		Discrete Cosine Transform (DCT)		Least Squares Linear Regression (LSLR)		Artificial Algae Algorithm (AAA)	
	Training MSE	Testing MSE	Training MSE	Testing MSE	Training MSE	Testing MSE	Training MSE	Testing MSE
NLR	$4.85 \times 10^{-6}$	$2.64 \times 10^{-5}$	$4.13 \times 10^{-5}$	$2.88 \times 10^{-5}$	$1.21 \times 10^{-6}$	$3.64 \times 10^{-5}$	$7.21 \times 10^{-7}$	$9.53 \times 10^{-6}$
LR	$3.62 \times 10^{-6}$	$4.79 \times 10^{-5}$	$6.92 \times 10^{-6}$	$1.35 \times 10^{-5}$	$7.72 \times 10^{-6}$	$1.96 \times 10^{-5}$	$6.98 \times 10^{-7}$	$4.23 \times 10^{-6}$
GMM	$6.13 \times 10^{-6}$	$2.26 \times 10^{-4}$	$7.63 \times 10^{-7}$	$9.22 \times 10^{-6}$	$4.57 \times 10^{-6}$	$1.39 \times 10^{-5}$	$3.81 \times 10^{-7}$	$4.52 \times 10^{-6}$
EM	<b><math>2.19 \times 10^{-7}</math></b>	<b><math>1.2 \times 10^{-6}</math></b>	$4.39 \times 10^{-6}$	$2.25 \times 10^{-5}$	$4.81 \times 10^{-6}$	$3.92 \times 10^{-5}$	$4.67 \times 10^{-7}$	$1 \times 10^{-5}$
BLDC	$4.47 \times 10^{-6}$	$6.56 \times 10^{-5}$	$7.94 \times 10^{-7}$	$5.8 \times 10^{-5}$	$3.72 \times 10^{-6}$	$1.56 \times 10^{-5}$	$3.52 \times 10^{-7}$	$3.97 \times 10^{-6}$
LoR	$3.24 \times 10^{-6}$	$2.26 \times 10^{-4}$	$3.32 \times 10^{-6}$	$1.09 \times 10^{-5}$	$8.37 \times 10^{-6}$	$2.26 \times 10^{-5}$	$7.61 \times 10^{-8}$	$3.82 \times 10^{-6}$
SDC	$9.62 \times 10^{-6}$	$2.31 \times 10^{-4}$	$9.13 \times 10^{-7}$	$4.62 \times 10^{-5}$	$4.87 \times 10^{-6}$	$1.52 \times 10^{-5}$	$9.93 \times 10^{-8}$	$3.84 \times 10^{-6}$
SVM (L)	$4.12 \times 10^{-5}$	$5.29 \times 10^{-4}$	$8.47 \times 10^{-7}$	$4.16 \times 10^{-6}$	$1.93 \times 10^{-8}$	$9.61 \times 10^{-6}$	$1.67 \times 10^{-8}$	$3.81 \times 10^{-6}$
SVM (Poly)	$6.41 \times 10^{-5}$	$2.34 \times 10^{-4}$	$2.19 \times 10^{-7}$	$6.41 \times 10^{-6}$	$5.77 \times 10^{-8}$	$1.24 \times 10^{-5}$	$1.62 \times 10^{-8}$	$2.05 \times 10^{-6}$
SVM (RBF)	$3.72 \times 10^{-7}$	$2.56 \times 10^{-5}$	<b><math>6.17 \times 10^{-8}</math></b>	<b><math>1.35 \times 10^{-6}</math></b>	<b><math>6.79 \times 10^{-9}</math></b>	<b><math>2.42 \times 10^{-6}</math></b>	<b><math>1.99 \times 10^{-10}</math></b>	<b><math>2.5 \times 10^{-8}</math></b>

Table 7 demonstrates the training and testing Mean Squared Error (MSE) performance of classifiers utilizing the Dragonfly Algorithm-based feature selection method across various dimensionality reduction techniques. The training MSE values range from  $10^{-6}$  to  $10^{-9}$ , while the testing MSE varies between  $10^{-5}$  and  $10^{-8}$ . The SVM (RBF) classifier, when combined with the Dragonfly feature selection method, achieved a minimal training MSE of  $1.66 \times 10^{-9}$  and a testing MSE of  $3.25 \times 10^{-8}$ . Notably, this feature selection method led to improvements in the training and testing performance of all classifiers. This enhancement is reflected in improved accuracy, MCC, and Kappa parameters, regardless of the specific dimensionality reduction technique employed.

**Table 7.** Analysis of MSE in classifiers for various DR techniques with Dragonfly feature selection methods.

Classifiers	Bessel Function		Discrete Cosine Transform (DCT)		Least Squares Linear Regression (LSLR)		Artificial Algae Algorithm (AAA)	
	Training MSE	Testing MSE	Training MSE	Testing MSE	Training MSE	Testing MSE	Training MSE	Testing MSE
NLR	$3.62 \times 10^{-6}$	$4.54 \times 10^{-5}$	$4.16 \times 10^{-6}$	$1.36 \times 10^{-5}$	$8.21 \times 10^{-6}$	$2.72 \times 10^{-5}$	$3.86 \times 10^{-6}$	$1.28 \times 10^{-5}$
LR	$4.36 \times 10^{-6}$	$7.12 \times 10^{-5}$	$2.84 \times 10^{-6}$	$1.39 \times 10^{-5}$	$9.4 \times 10^{-6}$	$3.8 \times 10^{-5}$	$2.51 \times 10^{-8}$	$4.32 \times 10^{-6}$
GMM	$7.58 \times 10^{-7}$	$4.71 \times 10^{-5}$	$5.66 \times 10^{-8}$	$7.84 \times 10^{-6}$	$3.61 \times 10^{-6}$	$2.09 \times 10^{-5}$	$4.63 \times 10^{-8}$	$1.02 \times 10^{-5}$
EM	$4.79 \times 10^{-7}$	$3.31 \times 10^{-5}$	$3.79 \times 10^{-8}$	$1.68 \times 10^{-5}$	$5.33 \times 10^{-6}$	$6.12 \times 10^{-5}$	$3.43 \times 10^{-8}$	$1.46 \times 10^{-5}$
BLDC	$6.52 \times 10^{-7}$	$4.16 \times 10^{-5}$	$2.92 \times 10^{-8}$	$4.49 \times 10^{-5}$	$7.54 \times 10^{-8}$	$9.12 \times 10^{-6}$	$7.68 \times 10^{-8}$	$8.1 \times 10^{-6}$
LoR	$6.54 \times 10^{-7}$	$5.04 \times 10^{-5}$	$7.23 \times 10^{-8}$	$6.05 \times 10^{-6}$	$1.92 \times 10^{-7}$	$2.23 \times 10^{-6}$	$4.84 \times 10^{-9}$	$3.36 \times 10^{-6}$
SDC	$3.86 \times 10^{-7}$	$2.57 \times 10^{-5}$	$8.95 \times 10^{-7}$	$3.08 \times 10^{-6}$	$7.52 \times 10^{-8}$	$6.31 \times 10^{-6}$	$1.63 \times 10^{-8}$	$2.52 \times 10^{-6}$
SVM (L)	$5.42 \times 10^{-7}$	$3.51 \times 10^{-5}$	$8.45 \times 10^{-7}$	$1.03 \times 10^{-5}$	$1.41 \times 10^{-7}$	$2.83 \times 10^{-5}$	$1.95 \times 10^{-7}$	$1.7 \times 10^{-6}$
SVM (Poly)	$9.67 \times 10^{-7}$	$7.23 \times 10^{-5}$	$6.67 \times 10^{-6}$	$7.08 \times 10^{-6}$	$6.3 \times 10^{-7}$	$1.05 \times 10^{-5}$	$6.42 \times 10^{-8}$	$5.33 \times 10^{-6}$
SVM (RBF)	<b><math>8.64 \times 10^{-8}</math></b>	<b><math>2.72 \times 10^{-6}</math></b>	<b><math>1.82 \times 10^{-8}</math></b>	<b><math>9.05 \times 10^{-7}</math></b>	<b><math>3.4 \times 10^{-8}</math></b>	<b><math>1.69 \times 10^{-6}</math></b>	<b><math>1.66 \times 10^{-8}</math></b>	<b><math>3.25 \times 10^{-8}</math></b>

7.2. Selection of Target

The non-diabetic class ( $T_{ND}$ ) target value is taken at the lower side from 0→1, and this is mapped according to the following constraint:

$$\frac{1}{N} \sum_{i=1}^N \mu_i \leq T_{ND} \tag{66}$$

Here,  $\mu_i$  represents the mean value of the input feature vectors for the  $N$  number of non-diabetic features considered for classification. Similarly, for the diabetic class ( $T_{Dia}$ ),

the target value is mapped to the upper end of the zero-to-one (0→1) scale. This mapping is established based on the following:

$$\frac{1}{M} \sum_{j=1}^N \mu_j \leq T_{Dia} \tag{67}$$

Here,  $\mu_j$  signifies the average value of input feature vectors for the  $M$  number of diabetic cases used for classification. It is important to highlight that the target value  $T_{Dia}$  is set to be higher than the average values of  $\mu_i$  and  $\mu_j$ . This selection of target values requires the discrepancy between them to be at least 0.5, as expressed by the following:

$$||T_{Dia} - T_{ND}|| \geq 0.5 \tag{68}$$

The targets for non-diabetic  $T_{ND}$  and diabetic  $T_{Dia}$  are chosen as 0.1 and 0.85, respectively. Once the target is fixed, MSE is used for evaluating the performance of the classifiers. Table 8 shows the selection of optimal parameters for the classifiers after training and testing process.

**Table 8.** Selection of optimal parametric values for classifiers.

Classifiers	Description
NLR	Uniform weight $w = 0.4$ , bias $b = 0.001$ , iteratively modified sum of least square error, criterion: MSE
Linear Regression	Uniform weight $w = 0.451$ , bias $b = 0.003$ , criterion: MSE
GMM	Mean covariance of the input samples and tuning parameter using EM steps. Criterion: MSE
EM	0.13 likelihood probability, 0.45 cluster probability, with convergence rate of 0.631. Condition: MSE
BDLC	$P(y)$ , prior probability: 0.5, class mean: 0.85, 0.1; criterion: MSE
Logistic regression	Threshold $H\theta(x) < 0.48$ with criterion: MSE
SDC	$\Gamma = 0.5$ along with mean of each class target values as 0.1 and 0.85
SVM (Linear)	C (Regularization Parameter): 0.85, class weights: 0.4, convergence criterion: MSE
SVM (Polynomial)	C: 0.76, coefficient of the kernel function (gamma): 10, class weights: 0.5, convergence criterion: MSE
SVM (RBF)	C: 1, coefficient of the kernel function (gamma): 100, class weights: 0.86, convergence criterion: MSE

### 8. Results and Findings

The study employs the conventional tenfold testing and training approach, where 10% of the input is dedicated to testing, while the remaining 90% is utilized for training. The selection of performance metrics is pivotal for assessing the efficacy of classifiers. The assessment of classifier performance, especially in binary classification scenarios like distinguishing between diabetic and non-diabetic cases from pancreatic microarray gene data, relies on the utilization of a confusion matrix. This matrix facilitates the computation of performance metrics including accuracy, F1 score, MCC, error rate, FM metrics, and Kappa, which are commonly utilized to gauge the comprehensive performance of the model. The relevant parameters associated with the classifiers for performance analysis are illustrated in Table 9.

The performance of the classifier was evaluated using several metrics, including Acc, F1 score, MCC, ER, FM, and Kappa. Accuracy is the fraction of predictions that are correct, and it is a measure of the overall performance of the classifier. F1 score is the harmonic mean of precision and recall, and it is a measure of the classifier’s ability to both correctly identify positive instances and to correctly identify negative instances. MCC is a measure of the correlation between the observed and predicted classifications, and it is a more sensitive metric than accuracy or F1 score. Error rate is the fraction of predictions that are incorrect, and it is the complement of accuracy. The FM metric is a generalization of the F-measure that adds a beta parameter, and it is a measure of the classifier’s ability to both correctly identify the values of positive and negative instances, with a weighting that can be adjusted to favor one or the other. Kappa is a statistic that measures agreement between observed and predicted classifications, adjusted for chance. The results are tabulated in Table 10.

**Table 9.** Performance metrics.

Metrics	Formula	Assessment Focus
Accuracy	$Acc = \frac{(TN+TP)}{(TN+FN+TP+FP)}$	Fraction of predictions that are correct
F1 Score	$F1 = \frac{2 \times TP}{(2 \times TP + FP + FN)}$	Harmonic mean of precision and recall
Matthews Correlation Coefficient (MCC)	$MCC = \frac{(TP \times TN - FP \times FN)}{\sqrt{(TP+FP) \times (TP+FN) \times (TN+FP) \times (TN+FN)}}$	Correlation between the observed and predicted classifications
Error Rate	$Error\ rate = \frac{(FP+FN)}{(TP+TN+FP+FN)}$	Fraction of predictions that are incorrect
FM Metric	$FM = \sqrt{\left(\frac{TP}{TP+FP}\right) \times \left(\frac{TP}{TP+FN}\right)}$	Generalization of the F-measure that adds a beta parameter
Kappa	$Kappa = \frac{(P_o - P_e)}{(1 - P_e)}$ $P_o = \frac{(TP+TN)}{(TP+TN+FP+FN)}$ $P_e = \frac{(TP+FP) \times (TP+FN) + (FP+TN) \times (FN+TN)}{(TP+TN+FP+FN)^2}$	Statistic that measures agreement between observed and predicted classifications, adjusted for chance

Abbreviations: *TP*—true positive: an accurate prediction where the true value was positive. *TN*—true negative: an accurate prediction where the true value was negative. *FP*—false positive: an inaccurate prediction where the actual value was negative. *FN*—false negative: an erroneous prediction where the actual value was positive.

**Table 10.** Parametric analysis of different classifiers through various DM techniques.

Dimensionality Reduction	Classifiers	Parameters					
		Accuracy (%)	F1 Score (%)	MCC	Error Rate (%)	FM (%)	Kappa
Bessel Function	NLR	54.2857	40.7407	0.0813	45.7142	42.1831	0.0743
	LR	58.5714	47.2727	0.1897	41.4285	49.1354	0.1714
	GMM	57.1428	48.2758	0.1995	42.8571	50.7833	0.1732
	EM	61.4285	54.2372	0.3092	38.5714	57.2892	0.2645
	BLDC	52.8571	40	0.0632	47.1428	41.5761	0.0571
	LoR	54.2857	40.7407	0.0813	45.7142	42.1831	0.0743
	SDC	54.2857	40.7407	0.0813	45.7142	42.1831	0.0743
	SVM (L)	52.8571	40	0.0632	47.1428	41.5761	0.0571
	SVM (Poly)	54.2857	42.8571	0.1084	45.7142	44.7214	0.0967
	SVM (RBF)	61.4285	52.6315	0.2805	38.5714	55.1411	0.2470
Discrete Cosine Transform (DCT)	NLR	75.7142	62.2222	0.4525	24.2857	62.6099	0.4465
	LR	71.4285	56.5217	0.3646	28.5714	57.0088	0.3577
	GMM	85.7142	76.1904	0.6617	14.2857	76.277	0.6601
	EM	80	69.5652	0.5609	20	70.1646	0.5504
	BLDC	65.7142	53.8461	0.3083	34.2857	55.3399	0.2881
	LoR	67.1428	59.6491	0.4072	32.8571	62.4932	0.3585
	SDC	81.4285	72.3404	0.6032	18.5714	73.1564	0.5882
	SVM (L)	70	60.3773	0.4162	30	62.2799	0.3849
	SVM (Poly)	72.8571	62.7451	0.4547	27.1428	64.2575	0.4291
	SVM (RBF)	88.5714	81.8181	0.7423	11.4285	82.1584	0.7358
Least Squares Linear Regression (LSLR)	NLR	67.1428	48.8888	0.2545	32.8571	49.1935	0.2511
	LR	65.7142	52	0.2829	34.2857	53.0723	0.2695
	GMM	82.8571	72.7272	0.6091	17.1428	73.0297	0.6037
	EM	72.8571	62.7451	0.4547	27.1428	64.2575	0.4291
	BLDC	62.8571	51.8518	0.2711	37.1428	53.6875	0.2479
	LoR	64.2857	57.6271	0.3728	35.7142	60.8698	0.3190
	SDC	75.7142	65.3061	0.4952	24.2857	66.4364	0.4757
	SVM (L)	64.2857	54.5454	0.3162	35.7142	56.6947	0.2857
	SVM (Poly)	71.4285	61.5384	0.4352	28.5714	63.2456	0.4067
	SVM (RBF)	84.2857	75.5555	0.6505	15.7142	76.0263	0.6418
Artificial Algae Algorithm (AAA)	NLR	80	66.6666	0.5254	20	66.7424	0.5242
	LR	80	68.1818	0.5424	20	68.4653	0.5377
	GMM	85.7142	77.2727	0.6757	14.2857	77.594	0.6698
	EM	84.2857	75.5555	0.6505	15.7142	76.0263	0.6418
	BLDC	78.5714	68.0851	0.5382	21.4285	68.853	0.5248
	LoR	77.1428	69.2307	0.5622	22.8571	71.1512	0.5254
	SDC	85.7142	78.2608	0.6918	14.2857	78.9352	0.6788
	SVM (L)	82.8571	75	0.6454	17.1428	76.0639	0.625
	SVM (Poly)	87.1428	80	0.7165	12.8571	80.4984	0.7069
	SVM (RBF)	90	84.4444	0.7825	10	84.9706	0.7720

Table 10 illustrates the performance analysis of ten classifiers, considering metrics such as Acc, F1 score, MCC, ER, F-measure, and Kappa values. This analysis is conducted for four DR methods without the incorporation of two feature selection methods. Table 9 reveals that the EM classifier in the Bessel function DR technique achieves a moderate accuracy of 61.42%, an F1 score of 54.23%, a moderate error rate of 38.57%, and an F-measure of 57.28%. However, the EM classifier exhibits a lower MCC value of 0.3092 and a Kappa value of 0.2645. On the other hand, the SVM (linear) classifier in the Bessel function DR method demonstrates a low accuracy of 52.85% along with a high error rate of 47.15%. Additionally, it exhibits an F1 score of 40% and an F-measure of 41.57%. The MCC and Kappa values for the SVM (linear) classifier are notably low, at 0.06324 and 0.05714, respectively. Across the Bessel function DR techniques, all classifiers exhibit poor performance in the various metrics. This trend can be attributed to the intrinsic properties of the Bessel function, which is evident from the non-negative values of the statistical parameters. Equally, the SVM (RBF) classifier in the context of the DCT DR technique achieves a respectable accuracy of 88.57%, complemented by a low error rate of 11.42%. Furthermore, it attains an F1 score of 81.81% and an F-measure of 82.15%. The MCC and Kappa values of the SVM (RBF) classifier reach 0.7423 and 0.7358, respectively. Within the SVM (RBF) classifier, the AAA DR technique exhibits a remarkable accuracy score of 90%, coupled with a low error rate of 10%. This is accompanied by an F1 score of 84.44% and an F-measure of 84.97%. The MCC and Kappa values of the SVM (RBF) classifier are noteworthy, totaling 0.7825 and 0.772, respectively. Remarkably, regardless of the DR technique employed, all classifiers manage to maintain accuracy within the range of 52% to 85%. This is primarily due to the inherent limitations of the DR techniques. Therefore, incorporating feature selection methods is highly recommended to enhance the performance of these classifiers.

Figure 13 provides an overview of the performance analysis of ten classifiers concerning the metrics of accuracy, F1 score, error rate, and F-measure values. This analysis is carried out within the context of four dimensionality reduction methods, specifically without feature selection methods. Table 10 shows that the EM classifier in the Bessel function DR technique achieves a modest accuracy of 61.42%, along with an F1 score of 54.23%. Moreover, it exhibits a moderate error rate of 38.57% and an F-measure of 57.28%. On the other hand, the SVM (linear) classifier in the Bessel function DR method demonstrates a lower accuracy of 52.85%. This classifier is accompanied by a higher error rate of 47.15%, an F1 score of 40%, and an F-measure of 41.57%. Across the performance metrics, all classifiers exhibit suboptimal performance within the Bessel function DR technique. This trend is observed consistently across various measures. However, the SVM (RBF) classifier within the DCT DR technique maintains an impressive accuracy level of 88.57%. Furthermore, it exhibits a commendably low error rate of 11.42%, an F1 score of 81.81%, and an F-measure of 82.15%. Employing the AAA DR technique in the SVM (RBF) classifier results in achieving an elevated accuracy rate of 90%. Additionally, this combination yields a notably low ER of 10% and an F1 score of 84.44%, accompanied by an F-measure of 84.97%.

Table 11 presents an in-depth analysis of the performance of ten classifiers concerning four DR methods integrated with the EHO feature selection technique. Notably, the SVM (RBF) classifier within the AAA DR technique achieves an exceptional accuracy of 95.71%. This classifier further demonstrates a commendable F1 score of 92.68%, accompanied by a notably low error rate of 4.28% and an impressive F-measure of 92.71%. Additionally, the SVM (RBF) has a high MCC value of 0.897 and a Kappa value of 0.8965. However, a contrasting performance is observed with the SVM(Linear) classifier within the Bessel function DR technique. Once again, this classifier registers a relatively low accuracy of 50%, coupled with a high error rate of 50%. Further metrics include an F1 score of 36.36% and an F-measure of 37.79%. Intriguingly, the SVM (Linear) classifier achieves null values for both MCC and Kappa, marking a unique and distinctive characteristic of its performance. All classifiers exhibit improved accuracy within the DCT, LSLR, and AAA DR techniques. However, the impact of the EHO feature selection method does not translate into substantial enhancements for classifiers employing the Bessel function DR method.

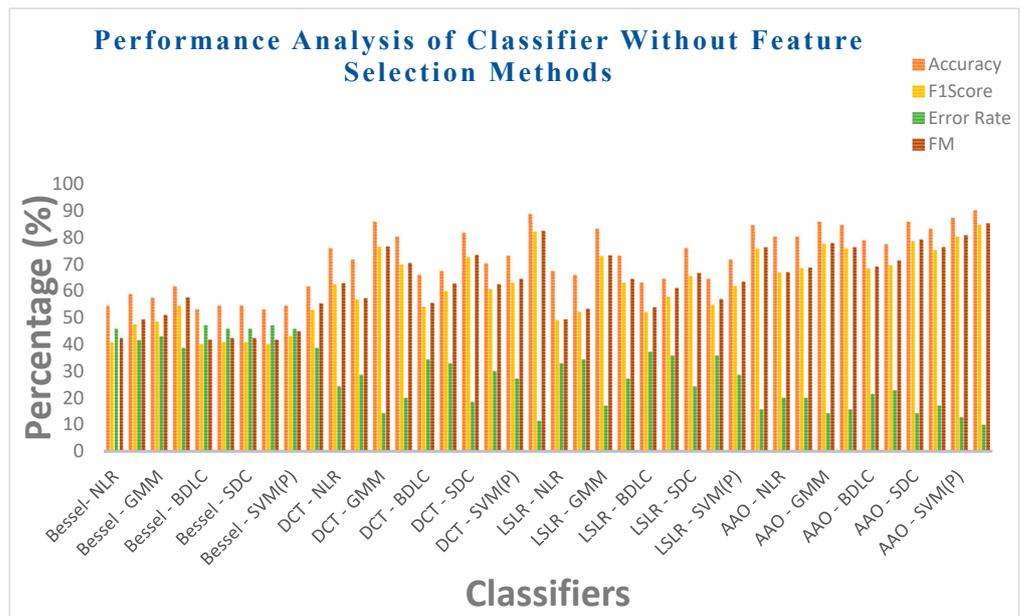


Figure 13. Different classifiers without feature selection methods.

Table 11. Performance metrics with Elephant Herding Optimization (EHO) feature selection method for different DR techniques.

Dimensionality Reduction	Classifiers	Parameters					
		Accuracy (%)	F1 Score (%)	MCC	Error Rate (%)	FM (%)	Kappa
Bessel Function	NLR	71.4285	61.5384	0.4352	28.5714	63.2456	0.4067
	LR	62.8571	50	0.2448	37.1428	51.387	0.2288
	GMM	54.2857	40.7407	0.0813	45.7142	42.1831	0.0743
	EM	81.4285	72.3404	0.6032	18.5714	73.1564	0.5882
	BLDC	60	46.1538	0.1813	40	47.4342	0.1694
	LoR	54.2857	40.7407	0.0813	45.7142	42.1831	0.0743
	SDC	54.2857	40.7407	0.0813	45.7142	42.1831	0.0743
	SVM (L)	50	36.3636	0	50	37.7964	0
	SVM (Poly)	52.8571	40	0.0632	47.1428	41.5761	0.0571
	SVM (RBF)	71.4285	60	0.4107	28.5714	61.2372	0.3913
Discrete Cosine Transform (DCT)	NLR	70	61.8181	0.4427	30	64.254	0.4
	LR	78.5714	68.0851	0.5382	21.4285	68.853	0.5248
	GMM	81.4285	72.3404	0.6032	18.5714	73.1564	0.5882
	EM	72.8571	64.1509	0.4796	27.1428	66.1724	0.4435
	BLDC	85.7142	77.2727	0.6757	14.2857	77.594	0.6698
	LoR	81.4285	71.1111	0.5845	18.5714	71.5542	0.5767
	SDC	85.7142	77.2727	0.6757	14.2857	77.594	0.6698
	SVM (L)	88.5714	80.9523	0.7298	11.4285	81.0443	0.7281
	SVM (Poly)	84.2857	75.5555	0.6505	15.7142	76.0263	0.6418
	SVM (RBF)	90	83.7209	0.7694	10	83.9254	0.7655
Least Squares Linear Regression (LSLR)	NLR	67.1428	59.6491	0.4072	32.8571	62.4932	0.3585
	LR	74.2857	64	0.4746	25.7142	65.3197	0.4521
	GMM	74.2857	65.3846	0.4987	25.7142	67.1984	0.4661
	EM	65.7142	57.1428	0.3615	34.2857	59.6285	0.3225
	BLDC	75.7142	65.3061	0.4952	24.2857	66.4364	0.4757
	LoR	72.8571	61.2244	0.4310	27.1428	62.2841	0.4140
	SDC	80	68.1818	0.5424	20	68.4653	0.5377
	SVM (L)	85.7142	76.1904	0.6617	14.2857	76.277	0.6601
	SVM (Poly)	80	69.5652	0.5609	20	70.1646	0.5504
	SVM (RBF)	88.5714	81.8181	0.7423	11.4285	82.1584	0.7358
Artificial Algae Algorithm (AAA)	NLR	81.4285	73.4693	0.6236	18.5714	74.7409	0.5991
	LR	87.1428	80	0.7165	12.8571	80.4984	0.7069
	GMM	85.7142	78.2608	0.6918	14.2857	78.9352	0.6788
	EM	81.4285	73.4693	0.6236	18.5714	74.7409	0.5991
	BLDC	87.1428	80	0.7165	12.8571	80.4984	0.7069
	LoR	87.1428	79.0697	0.7021	12.8571	79.2629	0.6985
	SDC	88.5714	80.9523	0.7298	11.4285	81.0443	0.7281
	SVM (L)	97.1428	94.7368	0.9302	2.85714	94.8683	0.9278
	SVM (Poly)	88.5714	81.8181	0.7423	11.4285	82.1584	0.7358
	SVM (RBF)	95.7142	92.6829	0.8970	4.28571	92.7105	0.8965

Figure 14 presents the analysis of the ten classifiers concerning the four DR methods combined with the EHO feature selection techniques. Furthermore, it is evident from the insights presented in Table 11 that the SVM (RBF) classifier, operating within the AAA DR technique, achieves an impressively high accuracy of 95.71%. Additionally, this classifier demonstrates a notable F1 score of 92.68%, accompanied by a commendably low error rate of 4.29% and an impressive F-measure of 92.71%. Equally, the SVM(Linear) classifier used within the Bessel function DR technique reflects a lower accuracy of 50%, coupled with a higher error rate of 50%. Correspondingly, the F1 score is registered at 36.36%, and the F-measure reaches 37.79%. Overall, the classifiers exhibit relatively low performance within the context of the Bessel function DR technique.

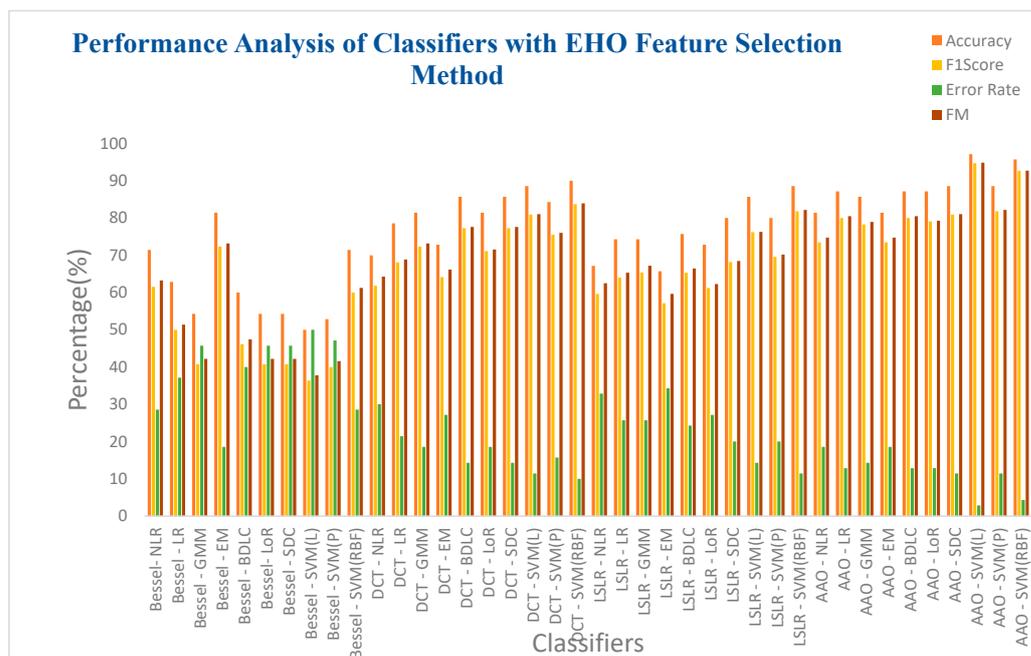


Figure 14. Different classifiers with EHO feature selection methods.

Table 12 presents the analysis of the ten classifiers concerning the four DR methods combined with the Dragonfly method. As depicted in Table 12, it is evident that the SVM (RBF), operating within the AAA DR technique, achieves an impressively high accuracy rate of 94.28%. Moreover, this classifier demonstrates a commendable F1 score of 90.47%, accompanied by a relatively low error rate of 5.72% and an appreciable F-measure of 90.57%. Furthermore, the SVM (RBF) classifier exhibits notable values of MCC and Kappa, standing at 0.866 and 0.864, respectively. On the other hand, the SVM (Polynomial) classifier, applied within the context of the Bessel function DR technique, achieves a lower accuracy rate of 58.57%. Correspondingly, it registers a higher error rate of 41.43%, along with an F1 score of 43.13% and an F-measure of 44.17%. However, the MCC and Kappa values for the SVM (Polynomial) classifier are notably lower, reaching 0.1364 and 0.1287, respectively. Among the classifiers utilized in the Bessel function DR method, only the SVM (RBF) classifier achieves an accuracy above 78%. Additionally, the SVM (RBF) classifier attains high accuracy in the DCT DR and LSLR DR methods, reaching 91% and 90%, respectively.

**Table 12.** Performance metric of different classifiers with four DR techniques with the Dragonfly feature selection method.

DR	Classifiers	Parameters					
		Accuracy (%)	F1 Score (%)	MCC	Error Rate (%)	FM (%)	Kappa
Bessel Function	NLR	64.2857	57.6271	0.3728	35.7142	60.8698	0.3190
	LR	60	44	0.1551	40	44.9073	0.1478
	GMM	64.2857	56.1403	0.3438	35.7142	58.8172	0.3027
	EM	78.5714	61.5384	0.4673	21.4285	61.5587	0.4670
	BLDC	65.7142	52	0.2829	34.2857	53.0723	0.2695
	LoR	64.2857	50.9803	0.2637	35.7142	52.2093	0.2489
	SDC	72.8571	59.5744	0.4083	27.1428	60.2464	0.3981
	SVM (L)	67.1428	56.6037	0.3529	32.8571	58.3874	0.3263
	SVM (Poly)	58.5714	43.1372	0.1364	41.4285	44.1771	0.1287
	SVM (RBF)	81.4285	66.6666	0.5384	18.5714	66.6886	0.5380
Discrete Cosine Transform (DCT)	NLR	80	68.1818	0.5424	20	68.4653	0.5377
	LR	78.5714	68.0851	0.5382	21.4285	68.853	0.5248
	GMM	84.2857	75.5555	0.6505	15.7142	76.0263	0.6418
	EM	74.2857	65.3846	0.4987	25.7142	67.1984	0.4661
	BLDC	85.7142	78.2608	0.6918	14.2857	78.9352	0.6788
	LoR	88.5714	80	0.72	11.4285	80	0.72
	SDC	88.5714	81.8181	0.7423	11.4285	82.1584	0.7358
	SVM (L)	82.8571	73.9130	0.6264	17.1428	74.5499	0.6146
	SVM (Poly)	84.2857	75.5555	0.6505	15.7142	76.0263	0.6418
	SVM (RBF)	91.4285	85.7142	0.7979	8.57142	85.8116	0.7961
Least Squares Linear Regression (LSLR)	NLR	75.7142	62.2222	0.4525	24.2857	62.6099	0.4465
	LR	65.7142	53.8461	0.3083	34.2857	55.3399	0.2881
	GMM	75.7142	62.2222	0.4525	24.2857	62.6099	0.4465
	EM	60	48.1481	0.2078	40	49.8527	0.1900
	BLDC	81.4285	72.3404	0.6032	18.5714	73.1564	0.5882
	LoR	80	65	0.51	20	65	0.51
	SDC	87.1428	79.0697	0.7021	12.8571	79.2629	0.6985
	SVM (L)	70	60.3773	0.4162	30	62.2799	0.3849
	SVM (Poly)	81.4285	72.3404	0.6032	18.5714	73.1564	0.5882
	SVM (RBF)	90	84.4444	0.7825	10	84.9706	0.7720
Artificial Algae Algorithm (AAA)	NLR	78.5714	68.0851	0.5382	21.4285	68.853	0.5248
	LR	87.1428	79.0697	0.7021	12.8571	79.2629	0.6985
	GMM	81.4285	73.4693	0.6236	18.5714	74.7409	0.5991
	EM	80	68.1818	0.5424	20	68.4653	0.5377
	BLDC	82.8571	75	0.6454	17.1428	76.0639	0.625
	LoR	88.5714	80.9523	0.7298	11.4285	81.0443	0.7281
	SDC	88.5714	81.8181	0.7423	11.4285	82.1584	0.7358
	SVM (L)	82.8571	73.9130	0.6264	17.1428	74.5499	0.6146
	SVM (Poly)	85.7142	78.2608	0.6918	14.2857	78.9352	0.6788
	SVM (RBF)	94.2857	90.4761	0.8660	5.71428	90.5789	0.8640

Figure 15 illustrates the performance assessment of the ten classifiers concerning the four DR methods, paired with the Dragonfly feature selection technique. It is observed from Table 12 that the SVM (RBF) classifier, within the AAA DR technique, attains a notably high accuracy rate of 94.28%. This classifier also demonstrates a commendable F1 score of 90.47%, coupled with a comparatively low ER of 5.72%, and a noteworthy F-measure of 90.57%. Conversely, the SVM (Polynomial) classifier, employed in the context of the Bessel function DR technique, registers a relatively low accuracy of 58.57%. Correspondingly, it records a higher error rate of 41.43%, accompanied by an F1 score of 43.13%, and an F-measure of 44.17%. Among the four dimensionality reduction methods, the SVM (RBF) classifier consistently achieves individual accuracy levels exceeding 81%. However, it is

important to note that the classifier’s performance in the Bessel function DR method, when paired with the Dragonfly feature selection, remains in the lower performance category.

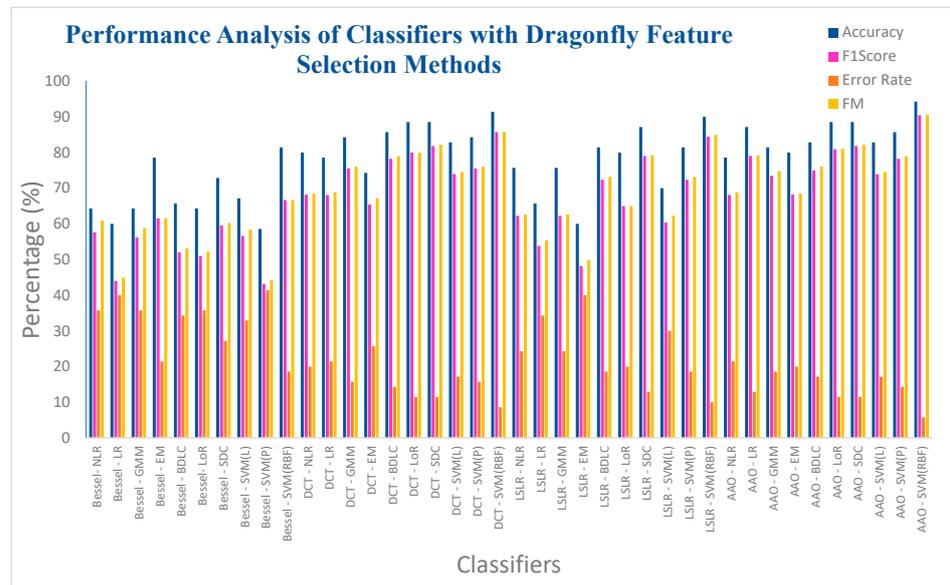


Figure 15. Different classifiers with Dragonfly feature selection method.

Figure 16 presents the comparative analysis of the MCC and Kappa parameters across the various classifiers concerning the four different DR techniques. This analysis was conducted for measuring the MCC and Kappa that serve as benchmarks, shedding light on the performance outcomes of the classifiers across diverse inputs. In this study, the inputs are categorized into three groups: dimensionally reduced without feature selection, with EHO feature selection, and with Dragonfly feature selection methods. These classifiers’ performance is evaluated based on the MCC and Kappa values derived from these inputs. The average MCC and Kappa values across the classifiers are calculated to be 0.2984 and 0.2849, respectively. A systematic approach is formulated to assess the classifiers’ performance, drawing insights from Figure 14. The MCC values are categorized into three ranges: 0.0–0.25, 0.251–0.54, and 0.55–0.9. Notably, the classifiers exhibit poor performance within the first range, while the MCC vs. Kappa slope demonstrates a significant upsurge within the second range of MCC values. In contrast, the third range of MCC values corresponds to a higher level of classifier performance, devoid of any substantial anomalies.

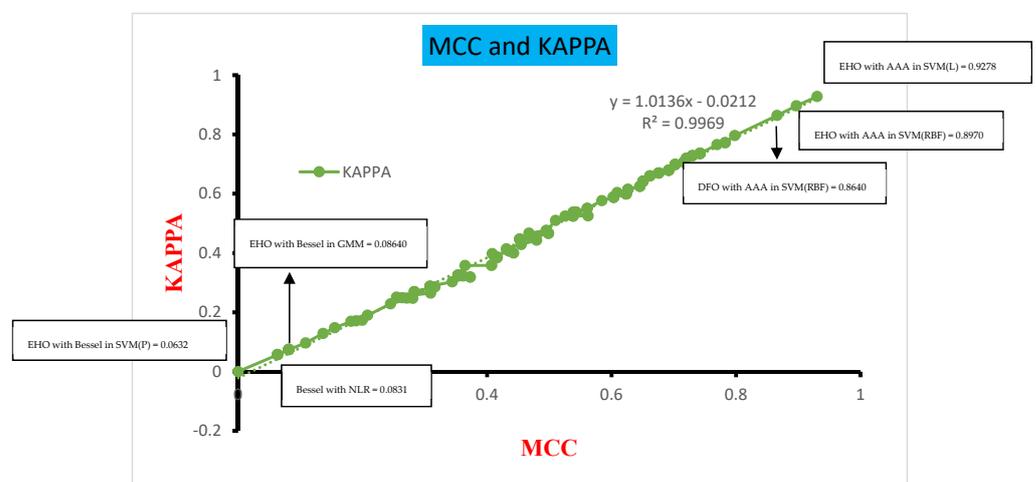
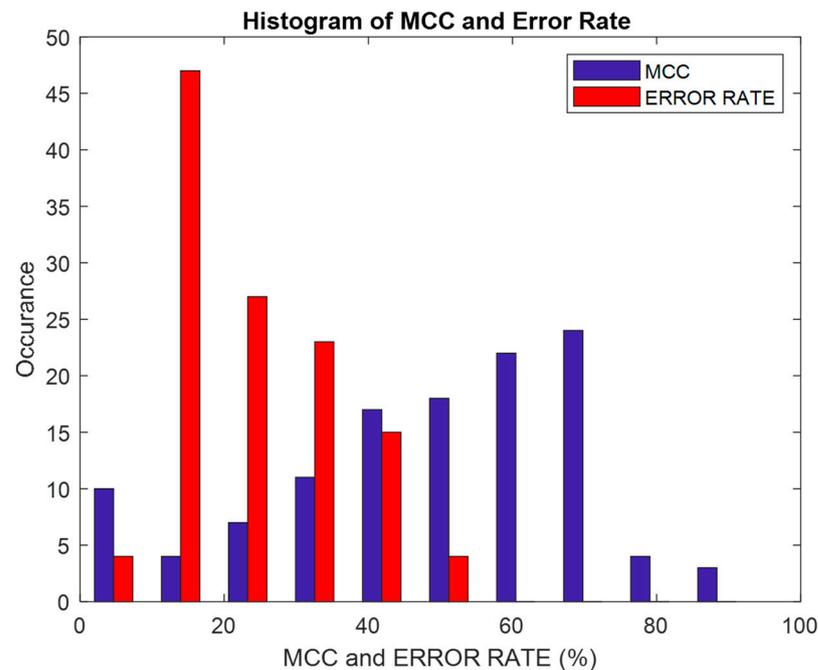


Figure 16. Classifier performance in terms of MCC and Kappa.

Figure 17 shows a histogram of the error rate and MCC (%) parameters that were analyzed. It can be seen that the maximum error rate is 50% and the maximum MCC is 90%. The histogram of the error rate is skewed at the right side of the graph, which indicates that for any of the DR methods, and irrespective of the feature selection method, the classifier's error rate does not go beyond 50%. The histogram of MCC depicts the classifier as being sparser at the edges and covering more points in the middle area.



**Figure 17.** Performance of error rate and MCC (%).

### 8.1. Computational Complexity (CC)

The analysis of the classifiers in this study considers their CC, which is determined based on the size of the input (denoted as  $O(n)$ ). A lower CC, indicated by  $O(1)$ , is desirable as it indicates that the complexity remains constant regardless of the input size. However, it is directly proportional to the number of inputs and computational complexity. It is notable that CC is independent of the size of the input, which is a favorable characteristic for any type of algorithm. If it increases logarithmically with the increase in 'n', it is represented as  $O(\log n)$ . Additionally, hybrid models of classifiers are used that incorporate DR techniques and feature selection methods in their classification process.

Table 13 presents the CC of the classifiers without incorporating feature selection methods. A noteworthy observation from the table is that the CC of all of the classifiers is relatively similar. However, their performance in terms of accuracy is relatively low. Among the classifiers, the Bessel function classifier demonstrates a moderate CC of  $O(n^3 \log n)$ , while the Discrete Cosine Transform, Least Squares Linear Regression, and Artificial Algae Algorithm exhibit higher CC with improved accuracy, represented by  $O(2n^4 \log 2n)$ ,  $O(2n^5 \log 4n)$ , and  $O(2n^5 \log 8n)$ , respectively, when compared to the other classifiers. Additionally, when considering the values of MCC and Kappa, the DST, LSLR, and AAA classifiers exhibit similar performance.

**Table 13.** CC of the classifiers without feature selection methods.

Classifiers	DR Method			
	Bessel Function	Discrete Cosine Transform (DST)	Least Squares Linear Regression (LSLR)	Artificial Algae Algorithm (AAA)
NLR	$O(n^2 \log n)$	$O(n^2 \log n)$	$O(n^3 \log 2n)$	$O(n^3 \log 4n)$
LR	$O(n^2 \log n)$	$O(n^2 \log n)$	$O(n^3 \log 2n)$	$O(n^3 \log 4n)$
GMM	$O(n^2 \log 2n)$	$O(n^2 \log 2n)$	$O(n^3 \log 2n)$	$O(n^3 \log 4n)$
EM	$O(n^3 \log n)$	$O(n^3 \log n)$	$O(n^3 \log 2n)$	$O(n^3 \log 4n)$
BLDC	$O(n^3 \log n)$	$O(n^3 \log n)$	$O(2n^3 \log 2n)$	$O(2n^3 \log 4n)$
LoR	$O(2n^2 \log n)$	$O(2n^2 \log n)$	$O(2n^4 \log 2n)$	$O(2n^4 \log 4n)$
SDC	$O(n^3 \log n)$	$O(n^3 \log n)$	$O(n^4 \log 2n)$	$O(n^4 \log 4n)$
SVM (L)	$O(2n^3 \log n)$	$O(2n^3 \log n)$	$O(2n^4 \log 2n)$	$O(2n^4 \log 4n)$
SVM (Poly)	$O(2n^3 \log 2n)$	$O(2n^3 \log 2n)$	$O(2n^4 \log 4n)$	$O(2n^4 \log 8n)$
SVM (RBF)	$O(2n^4 \log 2n)$	$O(2n^4 \log 2n)$	$O(2n^5 \log 4n)$	$O(2n^5 \log 8n)$

Table 14 illustrates the CC of the classifiers utilizing the EHO feature selection method. The table reveals that the CC of all of the classifiers is relatively similar, while their performance demonstrates significant accuracy. Similar to the case without feature selection, the Expectation Maximum classifier exhibits a higher computational complexity of  $O(n^5 \log n)$  along with remarkable accuracy. Regarding the DCT, LSLR, and AAA classifiers, they achieve similar CC to the SVM (RBF) classifier with  $O(2n^6 \log 2n)$ ,  $O(2n^7 \log 4n)$ , and  $O(2n^7 \log 8n)$ , respectively. Notably, the SVM (RBF) classifier in combination with the EHO feature selection technique for DCT, LSLR, and AAA achieves the highest accuracy among all classifiers, with accuracies of 90%, 88.57%, and 95.71%, respectively. Furthermore, the corresponding Kappa values for these classifiers are 0.7655, 0.65, and 0.8965, indicating their strong performance.

**Table 14.** CC of the classifiers with EHO feature selection method.

Classifiers	DR Method			
	Bessel Function	Discrete Cosine Transform (DCT)	Least Squares Linear Regression (LSLR)	Artificial Algae Algorithm (AAA)
NLR	$O(n^4 \log n)$	$O(n^4 \log n)$	$O(n^5 \log 2n)$	$O(n^5 \log 4n)$
LR	$O(n^4 \log n)$	$O(n^4 \log n)$	$O(n^5 \log 2n)$	$O(n^5 \log 4n)$
GMM	$O(n^4 \log 2n)$	$O(n^4 \log 2n)$	$O(n^5 \log 2n)$	$O(n^5 \log 4n)$
EM	$O(n^5 \log n)$	$O(n^5 \log n)$	$O(n^5 \log 2n)$	$O(n^5 \log 4n)$
BLDC	$O(n^5 \log n)$	$O(n^5 \log n)$	$O(2n^5 \log 2n)$	$O(2n^5 \log 4n)$
LoR	$O(2n^4 \log n)$	$O(2n^4 \log n)$	$O(2n^5 \log 2n)$	$O(2n^5 \log 4n)$
SDC	$O(n^5 \log n)$	$O(n^5 \log n)$	$O(n^6 \log 2n)$	$O(n^6 \log 4n)$
SVM (L)	$O(2n^5 \log n)$	$O(2n^5 \log n)$	$O(2n^6 \log 2n)$	$O(2n^6 \log 4n)$
SVM (Poly)	$O(2n^5 \log 2n)$	$O(2n^5 \log 2n)$	$O(2n^6 \log 4n)$	$O(2n^6 \log 8n)$
SVM (RBF)	$O(2n^6 \log 2n)$	$O(2n^6 \log 2n)$	$O(2n^7 \log 4n)$	$O(2n^7 \log 8n)$

Table 15 provides insights into the CC of the classifiers using the Dragonfly method. From the table, it can be seen that the CC of all of the classifiers is relatively similar, while their performance exhibits a significant level of accuracy. Notably, all four dimensionality reduction techniques demonstrate the highest CC compared to their counterparts. Specifically, the Bessel function, DCT, LSLR, and AAA classifiers achieve a computational complexity of  $O(8n^5 \log 2n)$ ,  $O(8n^5 \log 2n)$ ,  $O(8n^6 \log 4n)$ , and  $O(8n^6 \log 8n)$ , respectively. Regarding accuracy, the Bessel function, DCT, LSLR, and AAA classifiers achieve the highest accuracy values of 81.42%, 91.42%, 90%, and 95.71%, respectively. Moreover, the corresponding Kappa values for these classifiers are 0.538, 0.796, 0.772, and 0.864, indicating their robust performance. A comparison with previous work is provided in Table 16.

**Table 15.** CC of the classifiers with Dragonfly feature selection method.

Classifiers	DR Method			
	Bessel Function	Discrete Cosine Transform (DST)	Least Squares Linear Regression (LSLR)	Artificial Algae Algorithm (AAA)
NLR	$O(4n^3 \log n)$	$O(4n^3 \log n)$	$O(4n^4 \log 2n)$	$O(4n^4 \log 4n)$
LR	$O(4n^3 \log n)$	$O(4n^3 \log n)$	$O(4n^4 \log 2n)$	$O(4n^4 \log 4n)$
GMM	$O(4n^3 \log 2n)$	$O(4n^3 \log 2n)$	$O(4n^4 \log 2n)$	$O(4n^4 \log 4n)$
EM	$O(4n^4 \log n)$	$O(4n^4 \log n)$	$O(4n^4 \log 2n)$	$O(4n^4 \log 4n)$
BLDC	$O(4n^4 \log n)$	$O(4n^4 \log n)$	$O(8n^4 \log 2n)$	$O(8n^4 \log 4n)$
LoR	$O(8n^3 \log n)$	$O(8n^3 \log n)$	$O(8n^5 \log 2n)$	$O(8n^5 \log 4n)$
SDC	$O(4n^4 \log n)$	$O(4n^4 \log n)$	$O(4n^5 \log 2n)$	$O(4n^5 \log 4n)$
SVM (L)	$O(8n^4 \log n)$	$O(8n^4 \log n)$	$O(8n^5 \log 2n)$	$O(8n^5 \log 4n)$
SVM (Poly)	$O(8n^4 \log 2n)$	$O(8n^4 \log 2n)$	$O(8n^5 \log 4n)$	$O(8n^5 \log 8n)$
SVM (RBF)	$O(8n^5 \log 2n)$	$O(8n^5 \log 2n)$	$O(8n^6 \log 4n)$	$O(8n^6 \log 8n)$

**Table 16.** Comparison with previous work.

S.No	Author (with Year)	Description of the Population	Data Sampling	Machine Learning Parameter	Accuracy (%)
1.	Maniruzzaman et al., (2017) [53]	PIDD (Pima Indian diabetic dataset)	Cross-validation K2, K4, K5, K10, and JK	LDA, QDA, NB, GPC, SVM, ANN, AB, LoR, DT, RF	ACC: 92
2.	Pham et al., (2017) [54]	Diabetes: 12,000, aged between 18 and 100 Age (mean): 73	Training set—66%; tuning set 17%; test set—17%	RNN, CLST Memory (C-LSTM)	ACC—79
3.	Hertroijs et al., (2018) [55]	Total: 105,814 Age (mean): greater than 18	Training set of 90% and test set of 10% fivefold cross-validation	Latent Growth Mixture Modeling (LGMM)	ACC: 92.3
4.	ArellanoCampos et al., (2019) [56]	Base L: 7636 follow: 6144 diabetes: 331 age: 32–54	K = 10, cross-validation and bootstrapping model	Cox proportional hazard regression	ACC: 75
5.	Deo et al., (2019) [57]	Total: 140 diabetes: 14 imbalanced age: 12–90	Training set of 70% and 30% test set with fivefold cross-validation, holdout validation	BT, SVM (L)	ACC: 91
6.	Choi et al., (2019) [58]	Total: 8454 diabetes: 404 age: 40–72	Tenfold cross-validation	LoR, LDA, QDA, KNN	ACC: 78, 77, 76, 77
7.	Akula et al., (2019) [59]	PIDD Practice Fusion Dataset total: 10,000 age: 18–80	Training set: 800; test set: 10,000	KNN, SVM, DT, RF, GB, NN, NB	ACC: 86
8.	Xie et al., (2019) [60]	Total: 138,146 diabetes: 20,467 age: 30–80	Training set is around 67%, test set is around 33%	SVM, DT, LoR, RF, NN, NB	ACC: 81, 74, 81, 79, 82, 78
9.	Bernardini et al., (2020) [61]	Total: 252 diabetes: 252 age: 54–72	Tenfold cross-validation	Multiple instance learning boosting	ACC: 83
10.	Zhang et al., (2020) [62]	Total: 36,652 age: 18–79	Tenfold cross-validation	LoR, classification, and regression tree, GB, ANN, RF, SVM	ACC: 75, 80, 81, 74, 86, 76
11.	Jain et al., (2020) [63]	Control: 500 diabetes: 268 age: 21–81	Training set is around 70%, test set is around 30%	SVM, RF, k-NN	ACC: 74, 74, 76
12.	Kalagotla et al., (2021) [64]	Pima Indian dataset	Hold out k-fold cross-validation	Stacking multi-layer perceptron, SVM, LoR	ACC: 78
13.	Haneef et al., (2021) [65]	Total 44,659 age 18–69 data are imbalanced	Training set 80%, test set 20%	LDA	ACC: 67
14.	Deberneh et al., (2021) [66]	Total: 535,169, diabetes: 4.3% prediabetes: 36%, age: 18–108	Tenfold cross-validation	RF, SVM, XGBoost	ACC: 73, 73, 72
15.	Zhang et al., (2021) [67]	Total: 37,730, diabetes: 9.4% age: 50–70 imbalanced	Training set is around 80% test set is around 20% Tenfold cross-validation	Bagging boosting, GBT, RF, GBM	ACC: 82
16.	This article	Nordic Islet Transplantation program	Tenfold cross-validation	Bessel function, DCT, LSLR and AAA	95

LDA—Linear Discriminant Analysis; QDA—Quadratic Discriminant Analysis; NB—Naïve Bayes; GPC—Gaussian Process Classification; SVM—Support Vector Machine; ANN—Artificial Neural Network; AB—ADA Boost; LoR—Logistic Regression; DT—Decision Tree; RF—Random Forest; RNN—Recurrent Neural Network; CLST Memory—Convolutional Long Short-Term Memory; BT—Bagged Tree; KNN—k-Nearest Neighbor; GB—Gradient Boost; NN—Neural Network; k-NN—k-Nearest Neighbor; GBT—Bagging Boost GBT; ACC—accuracy.

As observed in Table 16, it is evident that a variety of machine learning classifiers, including SVM (RBF), NB, LoR, DT, NLR, RF, multilayer perceptron, and DNN, have been employed for diabetic classification using clinical databases. The accuracies of these classifiers span the range of 67% to 95%. However, the present investigation focuses on diabetes detection using microarray gene data, where SVM (RBF) stands out with an accuracy of 95%.

### 8.2. Limitations and Major Outcomes

The findings of this study may be limited to the specific population of type II diabetes mellitus patients and may not be applicable to other populations or different types of diabetes. The analysis in this study relies on microarray gene data, which may not be readily available or accessible in all healthcare settings. The methods proposed in this study, such as microarray gene arrays, may involve complex and expensive procedures that are not feasible for routine clinical practice. The performance of the classifiers in this study may be influenced by the presence of outliers in the data. Outliers can have a significant impact on the accuracy and reliability of the classification results. The developed classification approach, which utilizes various dimensionality reduction techniques and feature selection methods, has demonstrated its potential in effectively screening and predicting diabetic markers, while also identifying associated diseases such as strokes, kidney failure, and neuropathy. An outcome of this study is the establishment of a comprehensive database for the mass screening and sequencing of diabetic genomes. By incorporating microarray gene data and leveraging the proposed classification techniques, this database enables the identification of patterns and trends in diabetes outbreaks associated with different lifestyles.

The ability to detect diabetes in its early stages and predict associated diseases is of utmost importance for chronic diabetic patients. This will facilitate timely interventions, improve disease management, and, ultimately, lead to better patient outcomes. Overall, this study contributes valuable insights to the field and lays the foundation for further investigations into the early detection and management of type II diabetes mellitus patients.

## 9. Conclusions

The results showed that the classifiers exhibited lower accuracy and other performance metrics when using the BF-DR technique, which can be attributed to the inherent limitations of the Bessel function. However, the DCT and LSLR techniques produced improved accuracy and performance metrics for specific classifiers, such as the SVM (RBF) classifier. In particular, the AAA technique, combined with the SVM (RBF) classifier, achieved the highest accuracy of 90% without feature selection. The SVM (RBF) classifier in combination with the EHO feature selection technique achieved the highest accuracy values of 81.42, 90%, 88.57%, and 95.71% for BF, DCT, LSLR, and AAA, respectively. With the use of the Dragonfly feature selection method, which also showed promising results, the classifiers achieved high accuracy values of 81.42%, 91.42%, 90%, and 94.28% for BF, DCT, LSLR, and AAA, respectively. In terms of computational complexity, we observed that the classifiers exhibited similar complexities across the different dimensionality reduction techniques. However, their performance in terms of accuracy varied significantly. Notably, the SVM (RBF) classifier in combination with the EHO feature selection technique consistently achieved the highest accuracy values across the different dimensionality reduction techniques. In conclusion, this research article presents a novel method for detecting type II DM using microarray gene data. Future work will be carried out in the direction of the Convolution Neural Network (CNN), Deep Learning Network (DNN), LSTM, and hyperparameter tuning of classifiers. Moreover, this approach will be used for continuous monitoring in clinical practice.

**Author Contributions:** Conceptualization, D.C.; methodology, D.C. and H.R.; software, D.C.; validation, H.R.; formal analysis, D.C. and H.R.; investigation, D.C. and H.R.; resources, D.C. and H.R.; data curation, H.R.; writing—original draft, D.C.; writing—review and editing, H.R.; visualization, D.C.; supervision, H.R. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research received no external funding.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** The data that support the findings of this study are available from the corresponding author upon reasonable request.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Facts & Figures. International Diabetes Federation. Available online: <https://idf.org/about-diabetes/facts-figures/> (accessed on 20 August 2021).
2. Pradeepa, R.; Mohan, V. Epidemiology of type 2 diabetes in India. *Indian J. Ophthalmol.* **2021**, *69*, 2932–2938. [[CrossRef](#)]
3. Chockalingam, S.; Aluru, M.; Aluru, S. Microarray data processing techniques for genome-scale network inference from large public repositories. *Microarrays* **2016**, *5*, 23. [[CrossRef](#)]
4. Herman, W.H.; Ye, W.; Griffin, S.J.; Simmons, R.K.; Davies, M.J.; Khunti, K.; Wareham, N.J. Early detection and treatment of type 2 diabetes reduce cardiovascular morbidity and mortality: A simulation of the results of the Anglo-Danish-Dutch study of intensive treatment in people with screen-detected diabetes in primary care (ADDITION-Europe). *Diabetes Care* **2015**, *38*, 1449–1455. [[CrossRef](#)]
5. Strianese, O.; Rizzo, F.; Ciccarelli, M.; Galasso, G.; D’Agostino, Y.; Salvati, A.; Rusciano, M.R. Precision and personalized medicine: How genomic approach improves the management of cardiovascular and neurodegenerative disease. *Genes* **2020**, *11*, 747. [[CrossRef](#)] [[PubMed](#)]
6. Abul-Husn, N.S.; Kenny, E.E. Personalized medicine and the power of electronic health records. *Cell* **2019**, *177*, 58–69. [[CrossRef](#)] [[PubMed](#)]
7. Schnell, O.; Crocker, J.B.; Weng, J. Impact of HbA1c testing at point of care on diabetes management. *J. Diabetes Sci. Technol.* **2017**, *11*, 611–617. [[CrossRef](#)]
8. Lu, H.; Chen, J.; Yan, K.; Jin, Q.; Xue, Y.; Gao, Z. A hybrid feature selection algorithm for gene expression data classification. *Neurocomputing* **2017**, *256*, 56–62. [[CrossRef](#)]
9. American Diabetes Association Professional Practice Committee. 2. Classification and Diagnosis of Diabetes: Standards of Medical Care in Diabetes—2022. *Diabetes Care* **2022**, *45* (Suppl. S1), S17–S38. [[CrossRef](#)]
10. Jakka, A.; Jakka, V.R. Performance evaluation of machine learning models for diabetes prediction. *Int. J. Innov. Technol. Explor. Eng. Regul. Issue* **2019**, *8*, 1976–1980. [[CrossRef](#)]
11. Radja, M.; Emanuel, A.W.R. Performance evaluation of supervised machine learning algorithms using different data set sizes for diabetes prediction. In Proceedings of the 2019 5th International Conference on Science in Information Technology (ICSITech), Jogjakarta, Indonesia, 23–24 October 2019. [[CrossRef](#)]
12. Dinh, A.; Miertschin, S.; Young, A.; Mohanty, S.D. A data-driven approach to predicting diabetes and cardiovascular disease with machine learning. *BMC Med. Inform. Decis. Mak.* **2019**, *19*, 211. [[CrossRef](#)]
13. Yang, T.; Zhang, L.; Yi, L.; Feng, H.; Li, S.; Chen, H.; Zhu, J.; Zhao, J.; Zeng, Y.; Liu, H.; et al. Ensemble learning models based on noninvasive features for type 2 diabetes screening: Model development and validation. *JMIR Med. Inform.* **2020**, *8*, e15431. [[CrossRef](#)]
14. Muhammad, L.J.; Algehyne, E.A.; Usman, S.S. Predictive supervised machine learning models for diabetes mellitus. *SN Comput. Sci.* **2020**, *1*, 240. [[CrossRef](#)]
15. Kim, H.; Lim, D.H.; Kim, Y. Classification and prediction on the effects of nutritional intake on overweight/obesity, dyslipidemia, hypertension and type 2 diabetes mellitus using deep learning model: 4–7th Korea national health and nutrition examination survey. *Int. J. Environ. Res. Public Health* **2021**, *18*, 5597. [[CrossRef](#)] [[PubMed](#)]
16. Lawi, A.; Syarif, S. Performance evaluation of naive Bayes and support vector machine in type 2 diabetes Mellitus gene expression microarray data. In *Journal of Physics: Conference Series*; IOP Publishing: Bristol, UK, 2019; Volume 1341, p. 042018.
17. Ciaramella, A.; Staiano, A. On the role of clustering and visualization techniques in gene microarray data. *Algorithms* **2019**, *12*, 123. [[CrossRef](#)]
18. Velliangiri, S.; Alagumuthukrishnan, S.; Joseph, S.I.T. A review of dimensionality reduction techniques for efficient computation. *Procedia Comput. Sci.* **2019**, *165*, 104–111. [[CrossRef](#)]
19. Parand, K.; Nikarya, M. New numerical method based on generalized Bessel function to solve nonlinear Abel fractional differential equation of the first kind. *Nonlinear Eng.* **2019**, *8*, 438–448. [[CrossRef](#)]
20. Bell, W.W. *Special Functions for Scientists and Engineers*; Courier Corporation: North Chelmsford, MA, USA, 1967.

21. Kalaiyarasi, M.; Rajaguru, H. Performance Analysis of Ovarian Cancer Detection and Classification for Microarray Gene Data. *BioMed Res. Int.* **2022**, *2022*, 6750457. [[CrossRef](#)]
22. Ahmed, N.; Natarajan, T.; Rao, K.R. Discrete cosine transform. *IEEE Trans. Comput.* **1974**, *C-23*, 90–93. [[CrossRef](#)]
23. Epps, J.; Ambikairajah, E. Use of the discrete cosine transform for gene expression data analysis. In Proceedings of the Workshop on Genomic Signal Processing and Statistics, Baltimore, MD, USA, 26–27 May 2004; Volume 1.
24. Hotelling, H. Analysis of a complex of statistical variables into principal components. *J. Educ. Psychol.* **1933**, *24*, 417–441. [[CrossRef](#)]
25. Hastie, T.; Tibshirani, R.; Friedman, J.H.; Friedman, J.H. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*; Springer: New York, NY, USA, 2009; Volume 2.
26. Uymaz, S.A.; Tezel, G.; Yel, E. Artificial algae algorithm (AAA) for nonlinear global optimization. *Appl. Soft Comput.* **2015**, *31*, 153–171. [[CrossRef](#)]
27. Prabhakar, S.K.; Lee, S.W. An integrated approach for ovarian cancer classification with the application of stochastic optimization. *IEEE Access* **2020**, *8*, 127866–127882. [[CrossRef](#)]
28. Parhi, P.; Bisoi, R.; Dash, P.K. Influential gene selection from high-dimensional genomic data using a bio-inspired algorithm wrapped broad learning system. *IEEE Access* **2022**, *10*, 49219–49232. [[CrossRef](#)]
29. Ewees, A.A.; Al-Qaness, M.A.; Abualigah, L.; Algarni, Z.Y.; Oliva, D.; Yousri, D.; Elaziz, M.A. Enhanced feature selection technique using slime mould algorithm: A case study on chemical data. *Neural Comput. Appl.* **2023**, *35*, 3307–3324. [[CrossRef](#)] [[PubMed](#)]
30. Wang, G.G. Moth search algorithm: A bio-inspired metaheuristic algorithm for global optimization problems. *Memetic Comput.* **2018**, *10*, 151–164. [[CrossRef](#)]
31. Lin, Y.; Heidari, A.A.; Wang, S.; Chen, H.; Zhang, Y. An Enhanced Hunger Games Search Optimization with Application to Constrained Engineering Optimization Problems. *Biomimetics* **2023**, *8*, 441. [[CrossRef](#)]
32. Qiao, Z.; Li, L.; Zhao, X.; Liu, L.; Zhang, Q.; Hechmi, S.; Atri, M.; Li, X. An enhanced Runge Kutta boosted machine learning framework for medical diagnosis. *Comput. Biol. Med.* **2023**, *160*, 106949. [[CrossRef](#)] [[PubMed](#)]
33. He, X.; Shan, W.; Zhang, R.; Heidari, A.A.; Chen, H.; Zhang, Y. Improved Colony Predation Algorithm Optimized Convolutional Neural Networks for Electrocardiogram Signal Classification. *Biomimetics* **2023**, *8*, 268. [[CrossRef](#)] [[PubMed](#)]
34. Izci, D.; Ekin, S.; Eker, E.; Demirören, A. Biomedical application of a random learning and elite opposition-based weighted mean of vectors algorithm with pattern search mechanism. *J. Control. Autom. Electr. Syst.* **2023**, *34*, 333–343. [[CrossRef](#)]
35. Peng, L.; Cai, Z.; Heidari, A.A.; Zhang, L.; Chen, H. Hierarchical Harris hawks optimizer for feature selection. *J. Adv. Res.* **2023**. [[CrossRef](#)]
36. Su, H.; Zhao, D.; Heidari, A.A.; Liu, L.; Zhang, X.; Mafarja, M.; Chen, H. RIME: A physics-based optimization. *Neurocomputing* **2023**, *532*, 183–214. [[CrossRef](#)]
37. Wang, G.G.; Deb, S.; Coelho, L.D.S. Elephant herding optimization. In Proceedings of the 2015 3rd International Symposium on Computational and Business Intelligence (ISCBI), Bali, Indonesia, 9 December 2015; IEEE: Piscataway, NJ, USA, 2015; pp. 1–5.
38. Mirjalili, S. Dragonfly algorithm: A new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems. *Neural Comput. Appl.* **2016**, *27*, 1053–1073. [[CrossRef](#)]
39. Bharanidharan, N.; Rajaguru, H. Dementia MRI image classification using transformation technique based on elephant herding optimization with Randomized Adam method for updating the hyper-parameters. *Int. J. Imaging Syst. Technol.* **2021**, *31*, 1221–1245. [[CrossRef](#)]
40. Bharanidharan, N.; Rajaguru, H. Performance enhancement of swarm intelligence techniques in dementia classification using dragonfly-based hybrid algorithms. *Int. J. Imaging Syst. Technol.* **2020**, *30*, 57–74. [[CrossRef](#)]
41. Zhang, G.; Allaire, D.; Cagan, J. Reducing the Search Space for Global Minimum: A Focused Regions Identification Method for Least Squares Parameter Estimation in Nonlinear Models. *J. Comput. Inf. Sci. Eng.* **2023**, *23*, 021006. [[CrossRef](#)]
42. Draper, N.R.; Smith, H. *Applied Regression Analysis*; John Wiley & Sons: Hoboken, NJ, USA, 1998; Volume 326.
43. Llahi, O.; Rista, A. Prediction and Detection of Diabetes using Machine Learning. In Proceedings of the 20th International Conference on Real-Time Applications in Computer Science and Information Technology (RTA-CSIT), Tirana, Albania, 21–22 May 2021; pp. 94–102.
44. Prabhakar, S.K.; Rajaguru, H.; Lee, S.-W. A comprehensive analysis of alcoholic EEG signals with detrend fluctuation analysis and post classifiers. In Proceedings of the 2019 7th International Winter Conference on Brain-Computer Interface (BCI), Gangwon, Republic of Korea, 18–20 February 2019; IEEE: Piscataway, NJ, USA, 2019.
45. Liu, S.; Zhang, X.; Xu, L.; Ding, F. Expectation-maximization algorithm for bilinear systems by using the Rauch-Tung-Striebel smoother. *Automatica* **2022**, *142*, 110365. [[CrossRef](#)]
46. Zhou, W.; Liu, Y.; Yuan, Q.; Li, X. Epileptic seizure detection using lacunarity and Bayesian linear discriminant analysis in intracranial EEG. *IEEE Trans. Biomed. Eng.* **2013**, *60*, 3375–3381. [[CrossRef](#)]
47. Hamid, I.Y. Prediction of Type 2 Diabetes through Risk Factors using Binary Logistic Regression. *J. Al-Qadisiyah Comput. Sci. Math.* **2020**, *12*, 1–11. [[CrossRef](#)]
48. Adiwijaya, K.; Wisesty, U.N.; Lisnawati, E.; Aditsania, A.; Kusumo, D.S. Dimensionality reduction using principal component analysis for cancer detection based on microarray data classification. *J. Comput. Sci.* **2018**, *14*, 1521–1530. [[CrossRef](#)]

49. Zang, F.; Zhang, J.S. Softmax Discriminant Classifier. In Proceedings of the 3rd International Conference on Multimedia Information Networking and Security, Shanghai, China, 4–6 November 2011; pp. 16–20.
50. Yao, X.J.; Panaye, A.; Doucet, J.; Chen, H.; Zhang, R.; Fan, B.; Liu, M.; Hu, Z. Comparative classification study of toxicity mechanisms using support vector machines and radial basis function neural networks. *Anal. Chim. Acta* **2005**, *535*, 259–273. [[CrossRef](#)]
51. Fushiki, T. Estimation of prediction error by using K-fold cross-validation. *Stat. Comput.* **2011**, *21*, 137–146. [[CrossRef](#)]
52. Wang, Z.; Bovik, A.C. Mean squared error: Love it or leave it? A new look at signal fidelity measures. *IEEE Signal Process. Mag.* **2009**, *26*, 98–117. [[CrossRef](#)]
53. Maniruzzaman, M.; Kumar, N.; Abedin, M.M.; Islam, M.S.; Suri, H.S.; El-Baz, A.S.; Suri, J.S. Comparative approaches for classification of diabetes mellitus data: Machine learning paradigm. *Comput. Methods Programs Biomed.* **2017**, *152*, 23–34. [[CrossRef](#)] [[PubMed](#)]
54. Pham, T.; Tran, T.; Phung, D.; Venkatesh, S. Predicting healthcare trajectories from medical records: A deep learning approach. *J. Biomed. Inform.* **2017**, *69*, 218–229. [[CrossRef](#)] [[PubMed](#)]
55. Herttroijs, D.F.L.; Elissen, A.M.J.; Brouwers, M.C.G.J.; Schaper, N.C.; Köhler, S.; Popa, M.C.; Asteriadis, S.; Hendriks, S.H.; Bilo, H.J.; Ruwaard, D.; et al. A risk score including body mass index, glycated hemoglobin and triglycerides predicts future glycemic control in people with type 2 diabetes. *Diabetes Obes. Metab.* **2017**, *20*, 681–688. [[CrossRef](#)]
56. Arellano-Campos, O.; Gómez-Velasco, D.V.; Bello-Chavolla, O.Y.; Cruz-Bautista, I.; Melgarejo-Hernandez, M.A.; Muñoz-Hernandez, L.; Guillén, L.E.; Garduño-García, J.D.J.; Alvirde, U.; Ono-Yoshikawa, Y.; et al. Development and validation of a predictive model for incident type 2 diabetes in middleaged Mexican adults: The metabolic syndrome cohort. *BMC Endocr. Disord.* **2019**, *19*, 41. [[CrossRef](#)]
57. Deo, R.; Panigrahi, S. Performance assessment of machine learning based models for diabetes prediction. In Proceedings of the 2019 IEEE Healthcare Innovations and Point of Care Technologies, (HI-POCT), Bethesda, MD, USA, 20–22 November 2019. [[CrossRef](#)]
58. Choi, B.G.; Rha, S.-W.; Kim, S.W.; Kang, J.H.; Park, J.Y.; Noh, Y.-K. Machine learning for the prediction of new-onset diabetes mellitus during 5-year follow-up in non-diabetic patients with cardiovascular risks. *Yonsei Med. J.* **2019**, *60*, 191. [[CrossRef](#)]
59. Akula, R.; Nguyen, N.; Garibay, I. Supervised machine learning based ensemble model for accurate prediction of type 2 diabetes. In Proceedings of the 2019 Southeast Conference, Huntsville, AL, USA, 11–14 April 2019. [[CrossRef](#)]
60. Xie, Z.; Nikolayeva, O.; Luo, J.; Li, D. Building risk prediction models for type 2 diabetes using machine learning techniques. *Prev. Chronic Dis.* **2019**, *16*, E130. [[CrossRef](#)]
61. Bernardini, M.; Morettini, M.; Romeo, L.; Frontoni, E.; Burattini, L. Early temporal prediction of type 2 diabetes risk condition from a general practitioner electronic health record: A multiple instance boosting approach. *Artif. Intell. Med.* **2020**, *105*, 101847. [[CrossRef](#)]
62. Zhang, L.; Wang, Y.; Niu, M.; Wang, C.; Wang, Z. Machine learning for characterizing risk of type 2 diabetes mellitus in a rural Chinese population: The Henan rural cohort study. *Sci. Rep.* **2020**, *10*, 4406. [[CrossRef](#)]
63. Jain, S. A supervised model for diabetes divination. *Biosci. Biotechnol. Res. Commun.* **2020**, *13* (Suppl. S14), 315–318. [[CrossRef](#)]
64. Kalagotla, S.K.; Gangashetty, S.V.; Giridhar, K. A novel stacking technique for prediction of diabetes. *Comput. Biol. Med.* **2021**, *135*, 104554. [[CrossRef](#)]
65. Haneef, R.; Fuentes, S.; Fosse-Edorh, S.; Hrzic, R.; Kab, S.; Cosson, E.; Gallay, A. Use of artificial intelligence for public health surveillance: A case study to develop a machine learning-algorithm to estimate the incidence of diabetes mellitus in France. *Arch. Public Health* **2021**, *79*, 168. [[CrossRef](#)] [[PubMed](#)]
66. Deberneh, H.M.; Kim, I. Prediction of Type 2 diabetes based on machine learning algorithm. *Int. J. Environ. Res. Public Health* **2021**, *18*, 3317. [[CrossRef](#)] [[PubMed](#)]
67. Zhang, L.; Wang, Y.; Niu, M.; Wang, C.; Wang, Z. Nonlaboratory based risk assessment model for type 2 diabetes mellitus screening in Chinese rural population: A joint bagging boosting model. *IEEE J. Biomed. Health Inform.* **2021**, *25*, 4005–4016. [[CrossRef](#)] [[PubMed](#)]

**Disclaimer/Publisher’s Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.