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

Harikumar Rajaguru * and Dinesh Chellappan

Posted Date: 15 August 2023

doi: 10.20944/preprints202308.1033.v1

Keywords: Type II Diabetes Mellitus (DM), microarray gene data, Dimensionality Reduction (DR), Classification techniques, feature selection, LR, AAA, DOA, EHO

Article

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

Dinesh Chellappan [1] and Harikumar Rajaguru [2,*]

[1] Department of Electrical and Electronics Engineering, KPR Institute of Engineering and Technology, Coimbatore 641 407, Tamil Nadu – India. dinesh.chml@gmail.com

[2] Department of Electronics and Communication Engineering, Bannari Amman Institute of Technology, Sathyamangalam, 638 401, Tamil Nadu – India. harikumarrajaguru@gmail.com

* Correspondence: harikumarrajaguru@gmail.com

Abstract: Diabetes mellitus is a chronic disease that affects millions of people worldwide. Article focuses on detecting the presence of diabetes in patients using microarray gene data obtained from the pancreas. Handle the high-dimensional nature of the data, four different dimensionality reduction techniques, namely Bessel Function, Discrete Cosine Transform (DCT), Least Square Linear Regression (LSLR), and Artificial Algae Algorithm (AAA) are used. After reduced the data, Meta-heuristic algorithms are applied like Dragonfly Optimization Algorithm (DOA) and Elephant Herding Optimization Algorithm (EHO) for feature selection. Ten classification techniques are using to classify the data in both the format like without feature selection method and with feature selection method. The classification techniques are Non-Linear Regression (NLR), Linear Regression (LR), Gaussian Mixture Model (GMM), Expectation Maximum (EM), Bayesian Linear Discriminant Classifier (BLDC), Logistic Regression (LoR), Softmax Discriminant Classifier (SDC), Support Vector Machine (SVM) with linear kernel, Support Vector Machine (SVM) with polynomial kernel, and Support Vector Machine (SVM) with Radial Basis Function (RBF) kernel. Results showed that the AAA with SVM(RBF) achieved an accuracy of 90% without feature selection. However, when feature selection was applied, with EHO of AAA with SVM(RBF) exhibited the highest accuracy of 95.714%, followed closely by with DOA of AAA with SVM (RBF) at 94.28%.

Keywords: Type II Diabetes Mellitus (DM); microarray gene data; Dimensionality Reduction (DR); Classification techniques; feature selection; LR; AAA; DOA; EHO

1. Introduction

According to the International Diabetic Federation (IDF) Diabetes Atlas (2021), approximately 10.5% of the worlds adult population aged 20-79 years has diabetes, with almost half of them being unaware of their condition. 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 1 in 8 adults [1]. Type 2 II diabetes accounts for over 90% of all diabetes cases and is influenced by several factors, including socio-economic, demographic, environmental, and genetic factors. The rise in type 2 diabetes were attributed to urbanization, an aging population due to increase in life expectancy rate, declining physical activity levels, and increasing rates of overweight and obesity. Address the impact of diabetes, preventive measures for type 2 diabetes and early diagnosis and proper care for all types of diabetes are crucial. These interventions can help individuals with diabetes prevent or delay 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 diabetes in the country is rapidly escalating. In 2045, the number of adults living with diabetes in India could reach a staggering 134 million. Particularly the young age community of below 40 will be affected at Type II diabetes. Several risk

 doi:10.20944/preprints202308.1033.v1

factors like genetic predisposition, sedentary lifestyles, unhealthy dietary habits, obesity, urbanization, and mounting stress levels. India's Southern, urban, and northern regions exhibit higher rates compared to 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 $100 billion annually [4].

1.1. Genesis of Diagnosis of Diabetes Through Microarray Gene Technology

Developing accurate and efficient methods for detecting type II diabetes mellitus can lead to early detection and intervention. By analyzing microarray gene data, it becomes possible to identify specific genetic markers or patterns associated with diabetes[5]. This opens opportunities for personalized medicine, where treatment plans can be tailored based on an individual's genetic profile, leading to more targeted and effective interventions. Developing robust and reliable methods for diabetes detection from microarray gene data can be integrated into existing healthcare systems[6]. Exploring novel dimensionality reduction techniques, classification algorithms, and feature selection methods, as well as integrating other omics data to further enhance the accuracy and reliability of diabetes detection methods. The 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.

1.2. Review of literature

Diabetes mellitus (DM) is a chronic disease that affects millions of people worldwide. 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 for DM [9].Jakka et al [10] conducted a study for an experimental analysis using various machine learning classifiers, including K Nearest Neighbor (KNN), Decision Tree (DT), Naive Bayes (NB), Support Vector Machine (SVM), Logistic Regression (LR), and Random Forest (RF). The classifiers were trained and evaluated on the Pima Indians Diabetes dataset, which consists of nine attributes and is available on 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] gave a study aimed at evaluating the performance of various supervised classification algorithms for medical data analysis, specifically in disease diagnosis. The algorithms tested included Naive Bayes, Support Vector Machine (SVM), decision table, and J48. The evaluation utilized measurement variables such as Correctly Classified, Incorrect Classified, Precision, and Recall. The predictive database of diabetes was used as the testing dataset. The SVM algorithm demonstrated the highest accuracy of 77.3% among the tested algorithms, making it an effective tool for disease diagnosis. Dinh et al [12] analyzed to aim for assess the capabilities of machine learning models in identifying and predicting diabetes and cardiovascular diseases using survey data, including laboratory results. The National Health and Nutrition Examination Survey (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 gain of tree-based models. The ensemble model achieved an AU-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 AU-ROC score of 73.7% without laboratory data, and XGBoost performed the best with a score of 84.4% when using laboratory data. Key predictors for diabetes included waist size, age, self-reported weight, leg length, and sodium intake.

 doi:10.20944/preprints202308.1033.v1

Yang et al [13], conducted a study aimed to develop prediction models for diabetes screening using an ensemble learning approach. The dataset was obtained from the National Health and Nutrition Examination Survey (NHANES) from 2011-2016. Three simple machine learning methods (LDA, SVM and RF), the performance of the models was evaluated through 5-fold cross-validation and external validation using the Delong test. The study included 8057 observations and 12 attributes. The ensemble model based on linear discriminant analysis demonstrated the best performance, with an AUC of 0.849, an accuracy of 0.730, a sensitivity of 0.819, and a specificity of 0.709 in the validation set. Muhammed et al [14] gave a study utilized 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 of 88.76%. Kim et al [15] This study aimed to assess the impact of nutritional intake on overweight/obesity, dyslipidemia, hypertension, and type 2 diabetes mellitus (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 Korea National Health and Nutrition Examination Survey (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 overweight/obesity, dyslipidemia, hypertension, 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 Navie Bayes and SVM with Sigmoid kernel, were employed for classification, achieving accuracy rates of 88.89% and 83.33%, respectively. While numerous researchers have utilized the PIMA Indian diabetic dataset to classify and analyze diabetic and non-diabetic classes, only a limited number of studies have explored the microarray gene-based dataset for the identification of diabetic classes. Consequently, various performance metrics like Accuracy, Sensitivity, Specificity, and MCC were examined in the context of this microarray gene-based dataset.

2. Methodology

The methodology of the research is depicted in the Figure 1. It shows that four dimensionality reduction techniques such as Bessel Function (BF), Discrete Cosine Transform (DCT), Least Square Linear Regression (LSLR) are used. To analyse the data further, classification of data without feature selection and with feature selection methods are used. In this, with feature selection methods two optimization algorithm are used like Elephant Herding Optimization (EHO) and Dragon Fly Optimization Algorithm (DFO), Moreover, it has given to ten classifiers as NLR, LR, GMM, EM, BLDC, LoR, SDC, SVM -L, SVM-Poly and SVM-RBF to classify the classes as normal and diabetic.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 15 August 2023 doi:10.20944/preprints202308.1033.v1

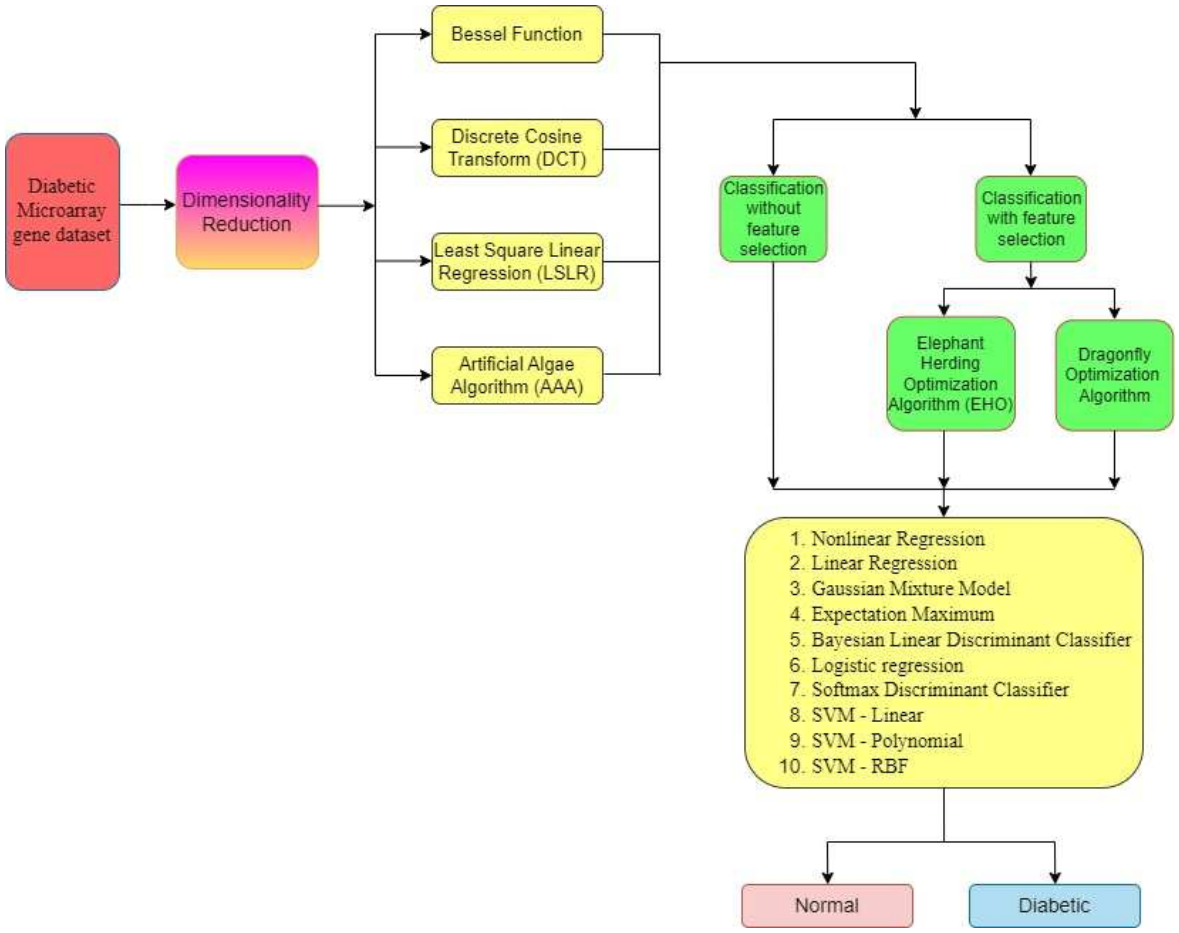

Figure 1. Diagrammatic Representation of the Workflow.

In Section 1, an introduction to the research is discussed. Section 2 presents the literature review. Section 3 the materials and methods are reviewed. Section 4 explains the process of feature extraction through dimensionality reduction techniques. In Section 5, feature selection methods are discussed, in Section 6 focuses classifiers are used. The results are presented in Section 7. Finding and conclusion are in the Section 8.

2.1. Role of Diabetic in Micro Array Gene:

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

3. Materials and Methods

Microarray gene data are readily available at many search engines. To our research, we obtained expression data from human pancreatic islets from the Nordic islet transplantation programme. (https://www.ncbi.nlm.nih.gov/bioproject/PRJNA178122). The data set included 28,735 genes from 57 non-diabetic and 20 diabetic cadaver donors. The data was preprocessed to select only 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 was 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 research suggest that microarray gene data can be used to develop effective

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 15 August 2023 doi:10.20944/preprints202308.1033.v1

methods for the detection of diabetes. The data is readily available and can be easily processed to identify the most relevant features for classification.

3.1. Data set

Focuses on utilizing microarray gene data to detect diabetes and explore the features associated with its secondary criteria using probability functions based on P-values. Additionally, we aimed to address the issue of false positive errors in the selection of significant genes. The data we used for our analysis is available through multiple portals and comprises a total of 28,735 human genes. We specifically considered 50 non-diabetic and 20 diabetic samples, 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 its informative content. The resulting dimensions were [2870*20] for the diabetic group and [2870*50] for the non-diabetic group. To further refine the dataset and improve classification accuracy, we applied feature selection techniques.

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

Table 1. Description of Pancreas Micro Array Gene Data set for Diabetic and Non-Diabetic Classes.

Data Set	No of Genes	Class 1 Diabetes	Class 2 Non-Diabetic	Total
Pancreas	28735	20	50	70

4. 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 of features (genes) increases, the complexity and computational cost 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 in mitigating the curse of dimensionality [18]. In high-dimensional spaces, data points tend to become sparse, leading to difficulties in accurately representing the underlying structure of the data.

4.1. Dimensionality Reduction

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

1. 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:

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

Where $\Gamma(\lambda)$ is the gamma function

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

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 plays a significant role in the analysis of the Bessel function.

(3)

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 15 August 2023 doi:10.20944/preprints202308.1033.v1

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

It is evident that the Bessel functions Jn(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 behaviour of Bessel functions in various mathematical contexts.

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

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

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

Lemma: 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 derivative operational matrix of Bessel functions can be derived using the following procedure: Consider the vector Jn = [J0(x), J1(x), J2(x), ..., Jn(x)]T, where J0, J1, J2, ..., Jn denote the Bessel functions of evaluated at x. To obtain the derivative operational matrix, we start with the derivative of J0(x) and denote it as J'0(x), where J'0(x) represents the derivative of J0(x) with respect to x. By constructing a matrix D, known as the derivative operational matrix, we can express Jn = DJ0, where D is a matrix that performs the differentiation operation on J0(x) to obtain Jn(x). This recursion relation allows for the efficient calculation and evaluation of Bessel functions, providing a valuable tool in various mathematical and scientific applications.

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

2. Discrete cosine Transform (DCT) as Dimensionality Reduction

The Discrete Cosine Transform (DCT) is a dimensionality reduction 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 represented as follows:

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

3. Least Square 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. LSLR applies PCA to transform high-dimensional data into a lower-dimensional space, upon which a linear regression model is applied. 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 independent variables (features) and the dependent variable (target). 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

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 15 August 2023 doi:10.20944/preprints202308.1033.v1

in the form (x1, y1), (x2, y2), ..., (xN, yN), where xi represents the i^{th} observation of the independent variables and yi corresponds to the observation of the target variable, the LSR solution can be represented as a linear equation:

$$z = \alpha_0 + \alpha_1 x_1 + \alpha_2 x_2 + .. + \alpha_p x_p \quad (10)$$

In the context of LSR, the linear model is characterized by the parameters ∝_1,∝_2,∝_3,…,∝_p, where p represents the number of independent variables. This minimization process is expressed through the following equation

$$SSE = \sum_{j=1}^{m} (z_i - (\alpha_0 + \alpha_1 x_1 + \alpha_2 x_2 + .. + \alpha_p x_p))^2 \quad (11)$$

After applying dimensionality reduction techniques to Microarray genes data, the resulting outputs are further analyzed using various statistical parameters such as mean, variance, skewness, kurtosis, Pearson correlation coefficient (PCC), f-test, T-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.

4. 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. The behavior of algal colony to act 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.

$$Population = \begin{bmatrix} X_{11} & X_{12} & \cdots & X_{1D} \\ X_{21} & X_{22} & \cdots & X_{2D} \\ \vdots & \vdots & \ddots & \vdots \\ X_{n1} & X_{n2} & \cdots & X_{nD} \end{bmatrix} \quad (12)$$

Where X_{nD} is algal cell in D^{th} dimension of n^{th} algal colony.

During the evolutionary process[27] of the Artificial Algae Algorithm (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.

$$H_i^t = \mu_i^t H_i^t \quad (13)$$

Where i = 1,2,3….N

H_i^t represents i^{th} algal colony in time t,

N represents no of algal colonies

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

$Maximum^t$= max H_i^t , whereas i = 1,2,3…N

$Minimum^t$ = min H_i^t , Whereas i = 1,2,3… N

$Minimum^t = Maximum^t$, whereas m = 1,2,3…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.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 15 August 2023 doi:10.20944/preprints202308.1033.v1

Adaptation occurs in the AAA when algal colonies that cannot grow sufficiently in their environment attempt to resemble the largest colony. This process leads to a change in the starvation level within the algorithm. Each artificial alga starts with a starvation value of zero, which increases over time when the algal cell receives insufficient light. The artificial alga with the highest starvation value becomes the focus of adaptation.

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

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

Helical Movement: Algal cells and colonies exhibit specific swimming behaviour, 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 modelled as a spherical shape, with its size determined by its volume, and the friction surface is equivalent to the surface area of a hemisphere,

$$\tau(x_i) = 2\pi r^2 \tag{15}$$

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

where $\tau(x_i)$ is the friction surface.

The helical movement of algal cells involves three randomly determined dimensions. One of these dimensions corresponds to linear movement, as described by Equation (18), while the other two dimensions contribute to angular movement, as expressed by Equations (19) and (20). Equation (18) is utilized for one-dimensional problems, enabling the algal cell or colony to move in a single direction. In two-dimensional problems, algal movement follows a sinusoidal pattern, and therefore, Equations (18) and (20) are employed. For three or more dimensions, algal movement takes on a helical trajectory, and Equations (18)-(20) are utilized. The step size of the movement is determined by the friction surface and the distance to the source of light.

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

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

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

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

The variables α and β are in the range [0, 2π], while p is within the interval [-1, 1]. Δ represents the shear force, and $\tau^t(X_i)$ denotes the friction surface area of the i-th algal cell.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 15 August 2023 doi:10.20944/preprints202308.1033.v1

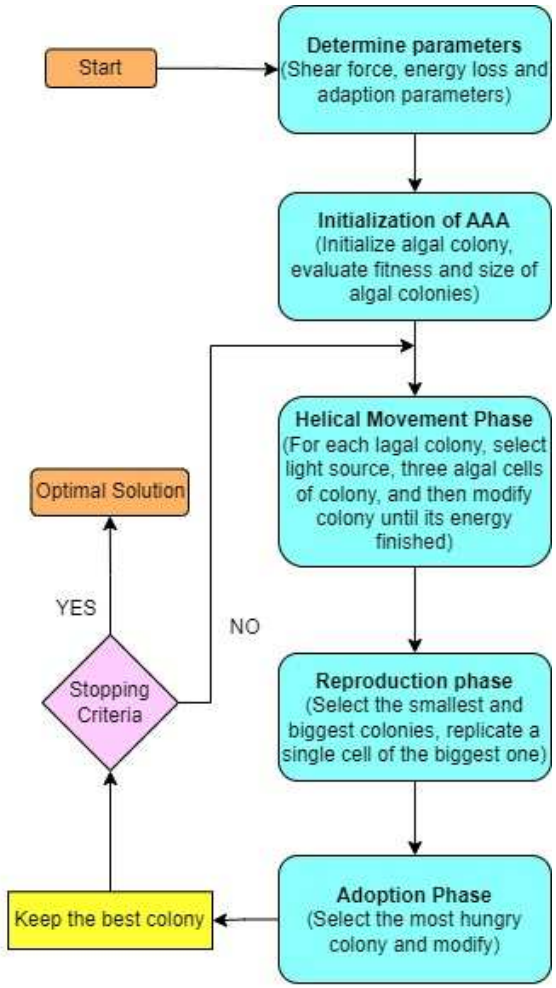

Figure 2. Flow diagram for Artificial Algae Algorithm.

4.2. Statistical Analysis

The dimensionally reduced Micro array genes through four DR methods are analysed by the statistical parameters like mean, variance, skewness, kurtosis, Pearson correlation coefficient (PCC), and CCA to identify whether the outcomes are representing the underlying micro array genes properties in the reduced subspace. As mentioned in the Table 2 that in the AAA based DR methods depicts higher values of mean, and variance among the classes. All the other three DR Methods namely, Bessel Function, DCT, and LSLR display low and overlapping values of mean and variance among the classes. The negative skewness depicted only by the LSLR DR method which indicates the presence of skewed components embedded in the classes. DCT and LSLR DR methods indicate negative kurtosis. This in turn leads to the observance that the DR methods are not modifying the underlying Micro array genes characteristics. PCC values indicate the high correlation within the class of attained outputs. In the case of Bessel function DR Method all the four statistical parameters are positive and minimum values. This indicates that the statistical parameters are associated with non-Gaussian and non-linear one. The same is further examined by the histogram, Normal probability plots and Scatter plots of DR techniques outputs. Canonical correlation Analysis (CCA) visualizes the correlation of DR methods outcomes among the Diabetic and non-Diabetic cases. The low CCA value in the Table 2 indicates that the DR outcomes are less correlated among the two classes.

Table 2. Statistical Analysis for Different Dimensionality Reduction Techniques.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 15 August 2023 doi:10.20944/preprints202308.1033.v1

Statistical Parameters	Bessel Function		Discrete Cosine Transform (DCT)		Least Square Linear Regression (LSLR)		Artificial Algae Algorithm (AAA)	
	DP	NDP	DP	NDP	DP	NDP	DP	NDP
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	0.00041 7	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 the Histogram of Bessel Function DR Techniques in Diabetic Gene Class. It is identified that in the Figure 3 the histogram depicts a skewed group of values, gap, and presence of non-linearity in this method. The patient from 1 to 10 are represented as x(:,1) to x(:,10).

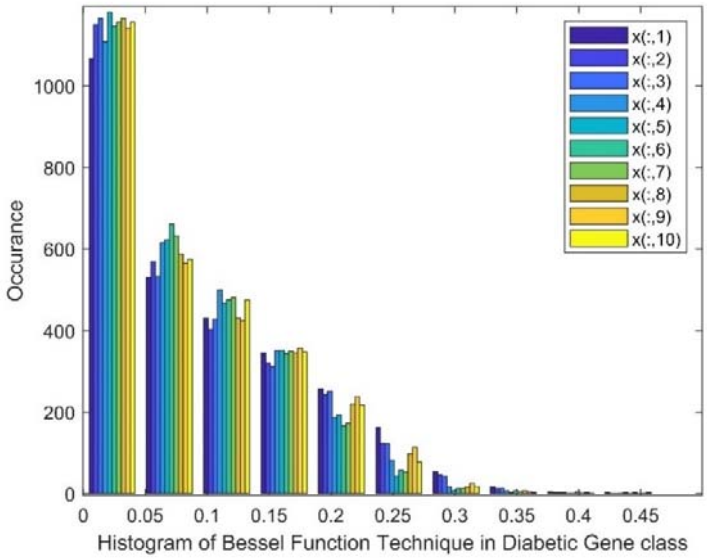

Figure 3. Histogram of Bessel Function Techniques in Diabetic Gene Class.

Figure 4 exhibits the Histogram of Bessel function DR Techniques in non-Diabetic Gene Class with the marker of x(:,1) represents as patient 1 and x(:,10) represents as patient 10. It is observed from the Figure 4 that the histogram depicts a skewed group of values, gap and presence of non-linearity in this method.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 15 August 2023 doi:10.20944/preprints202308.1033.v1

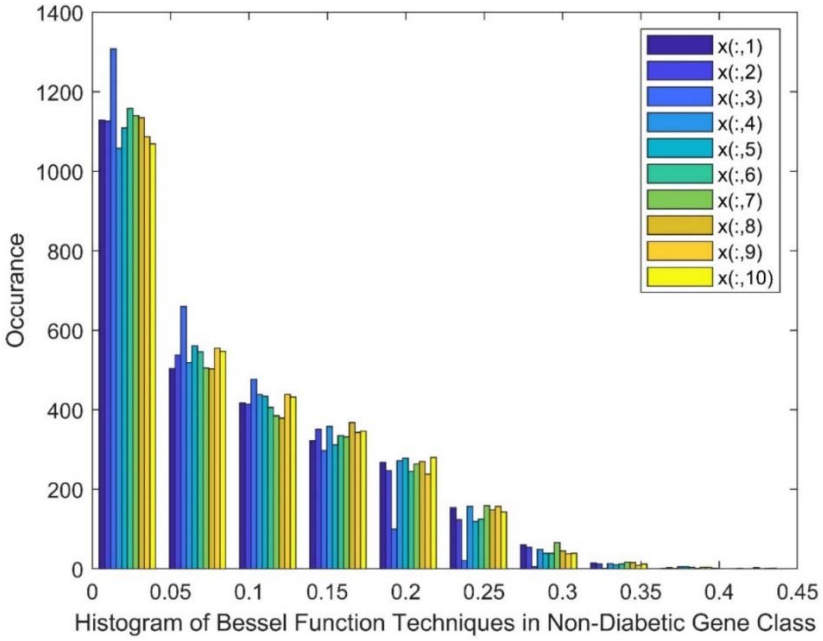

Figure 4. Histogram of Bessel Function Techniques in Non-Diabetic Gene Class.

In this Figure 5, Data 1 to 5 represents references, data 6 to 10 notices upper bound and data 11 to 15 cluster variable points. It indicates the Normal Probability plot for DCT DR Techniques Features for Diabetic Gene Class. As exhibits from the figure 5 that the normal probability plot displays the total cluster of DCT DR outputs and the presence of non-linearly correlated variables among the classes.

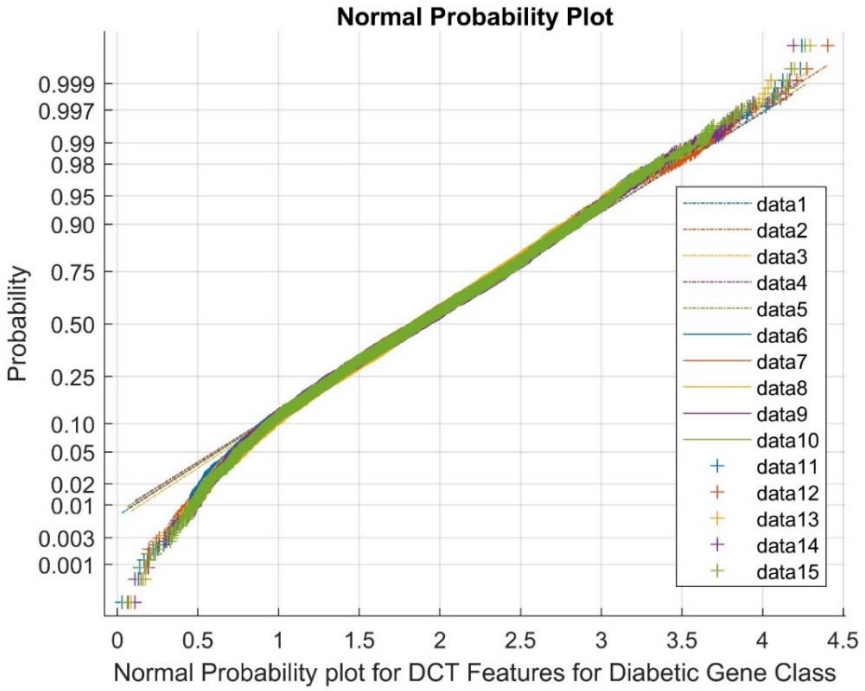

Figure 5. Normal Probability plot for DCT Features for Diabetic Gene Class.

Figure 6 shows the normal Probability plot for DCT DR Techniques Features for non-Diabetic Gene Class as data is presented from 1 to 5 as references, and upper bound value are represented from 6 to 10, and cluster variable points from 11 to 15. As exhibits from the figure 6 that the normal

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 15 August 2023 doi:10.20944/preprints202308.1033.v1

probability plot displays the total cluster of DCT DR outputs and the presence of non-linearly correlated variables among the classes. This is due to low values of mean and variance and the presence of negative kurtosis variables in the DR method outcomes.

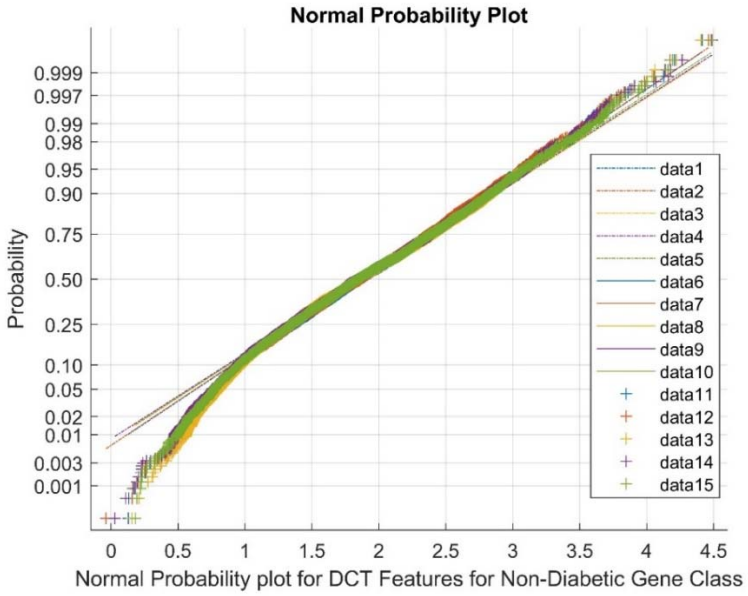

Figure 6. Normal Probability plot for DCT Features for Non-Diabetic Gene Class.

Data 1 to 5 are mentioned in the Figure 7 as references, data 6 to 10 are mentioned as upper bound values and data 11 to 15 are mentioned as variable points. It is normal Probability plot for LSLR DR Techniques Features for Diabetic Gene Class. As mentioned by the figure 6 that the normal probability plots display the group clusters for LSLR DR outputs. This indicates the presence of non-Gaussian and non-linearly variables within the classes. This is due to low variance and negative Kurtosis variables of the DR method outcomes.

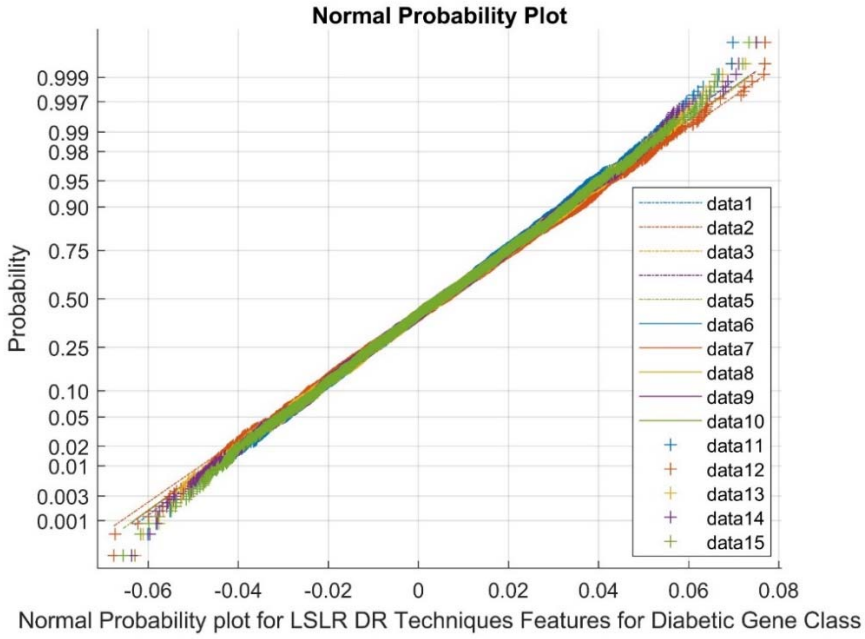

Figure 7. Normal Probability plot for Least Square Linear Regression DR Techniques Features for Diabetic Gene Class.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 15 August 2023 doi:10.20944/preprints202308.1033.v1

Figure 8 demonstrates the normal Probability plot for LSLR DR Techniques Features for non-Diabetic Gene Class. As shown in the figure 8 that the normal probability plots display the discrete group of clusters for LSLR DR outputs. Data representation from 1 to 5 as references, from 6 to 10 as upper bound and from 11 to 15 as variable points. This indicates the presence of non-Gaussian and nonlinear variables within the classes. This is due to low variance and flat Kurtosis variables of the DR method outcomes.

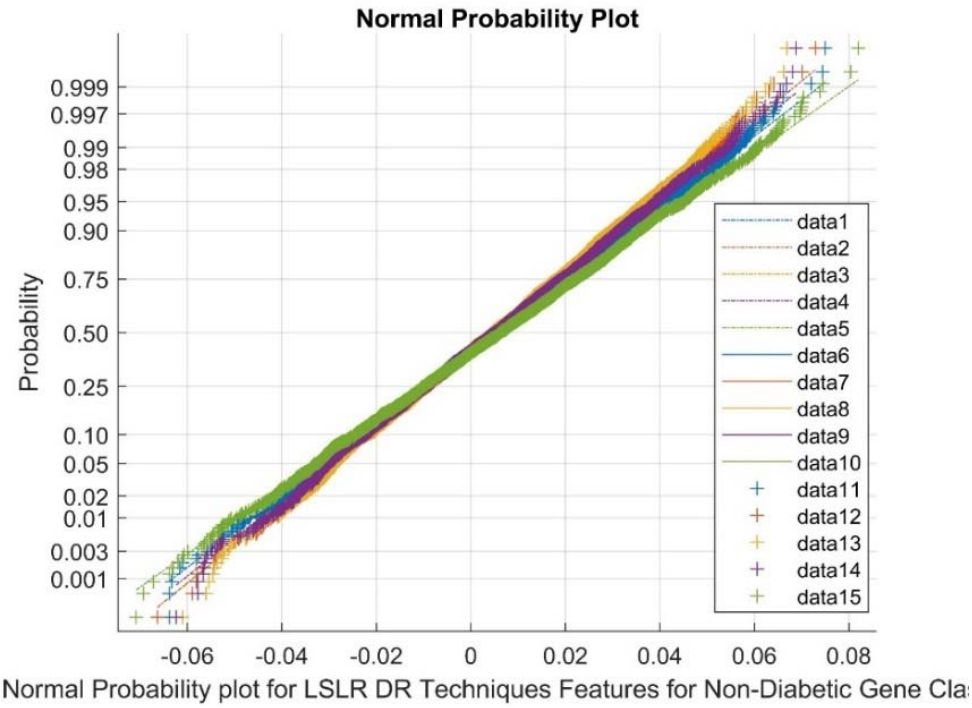

Figure 8. Normal Probability plot for Least Square Linear Regression DR Techniques Features for Non-Diabetic Gene Class.

Figure 9 shows the Scatter plot for AAA Algorithm DR Techniques Features for Diabetic and non-Diabetic Gene Class. As revealed by the figure 9 that the scatter plots from AAA algorithm displays the total clustering and overlapping of the variables of both classes across the entire subspace. The scatter plot also indicates the presence of non-Gaussian, non-linear and higher values of all statistical parameters. Furthermore, the AAA algorithm will have heavy computational cost on the classifier design. To reduce the burden of the classifiers, a feature selection process which comprises the elephant Herd Optimization (EHO) and Dragon fly algorithms are initiated

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 15 August 2023 doi:10.20944/preprints202308.1033.v1

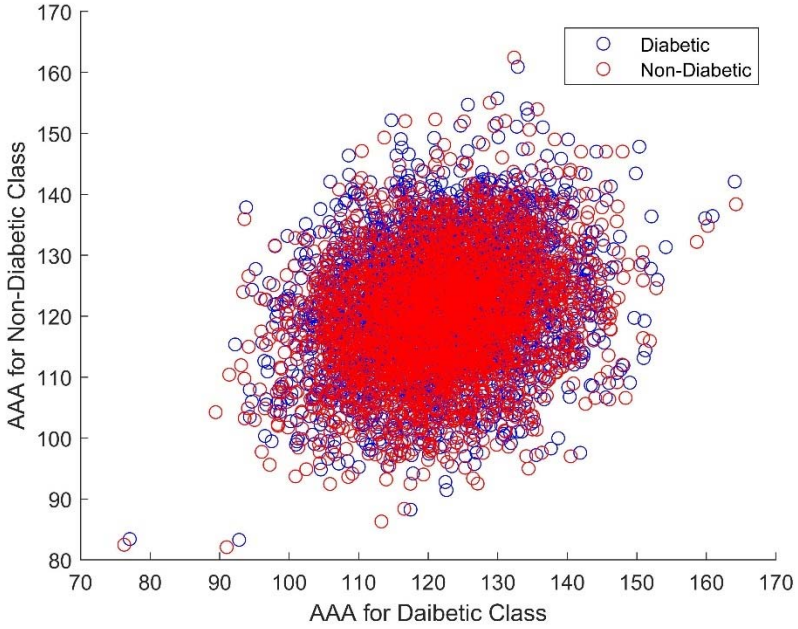

Figure 9. Scatter plot for AAA DR Technique with Diabetic and Non-Diabetic Gene classes.

5. Feature Selection Methods

Two metaheuristic algorithms are used in this article are the EHO algorithm and DOA. By emulating the behaviour observed in elephants and dragonflies, respectively, these algorithms offer effective approaches to address the challenges in optimization.

1. Elephant Herding Optimization (EHO) algorithm

Elephant Herding Optimization (EHO) is a meta-heuristic algorithm introduced by Wang et al. [28], inspired by the behaviour 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. 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.

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

The EHO algorithm [29] involves updating the positions of elephants within the herd. This update process considers both the old position ($yiold$) and the new position ($yinew$) of each elephant. A control parameter (α), which falls within the range of [0,1], is used in conjunction with a randomly generated number (r ∈ [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 ($Ybest$) for each elephant within the herd. This best position is determined by considering the control parameter (β), which falls within the

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 15 August 2023 doi:10.20944/preprints202308.1033.v1

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.

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

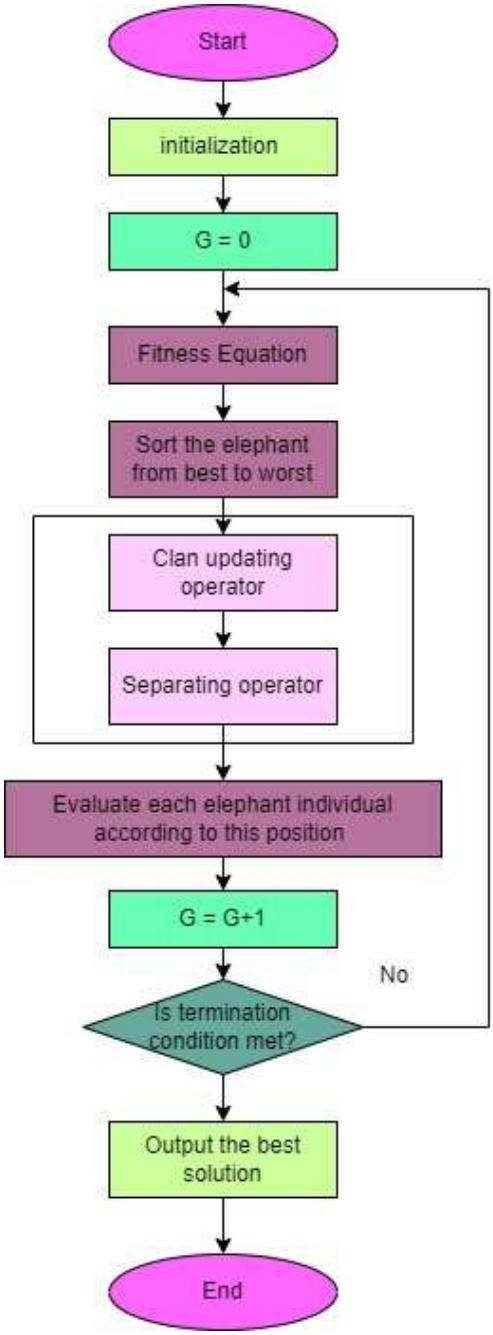

Figure 10. Flow Chart of Elephant Herding Optimization Algorithm.

2. 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 [30], this algorithm mimics the dynamic and static swarming behaviors observed in nature. During the dynamic or exploitation phase, dragonflies form large swarms and travel in a specific direction to confuse potential threats. In the static or exploration phase, they form smaller groups, moving within a limited area to hunt and attract prey [31]. The DA is guided by five fundamental principles:

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 15 August 2023 doi:10.20944/preprints202308.1033.v1

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 Ki denote the current position and the ith position of a dragonfly, respectively, while N represents the total number of neighboring flies.

1. 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 ith individual aimed at maintaining separation from other dragonflies.

2. 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 represents the velocity of the ith individual.

3. Cohesiveness: This represents the inclination of individual flies to converge towards the centre of swarms. The calculation for this is given by.

$$Co_j = \frac{\sum_{i=1}^{n} k_i}{N} - k \tag{26}$$

4. Attraction: The attraction towards the food source is quantified as

$$H_j = K^+ - K \tag{27}$$

The attraction towards the food source is represented by H_j, while K^+ represents the position of the food source.

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

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

The calculation of the outward distance determines the diversion from the enemy, and it is expressed as.

The step vector (ΔK) and the current position vector (K) are used to update the locations of artificial dragonflies within the search space. The movement direction of the dragonfly is determined by the step vector (ΔK), which is calculated as follows:

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

The separation weight (s), alignment weight (a), cohesion weight (c), attraction weight (h), and enemy weight (d) influence the behaviour of the dragonfly algorithm. The inertia weight is represented by "ω," and "t" represents the iteration number. By adjusting these weights, the exploration and exploitation phases of the algorithm can be achieved. The position of the ith dragonfly at t + 1 iterations is determined by the following equation:

$$K_j^{t+1} = K_j^t + \Delta K_j^{t+1} \tag{9}$$

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 15 August 2023 doi:10.20944/preprints202308.1033.v1

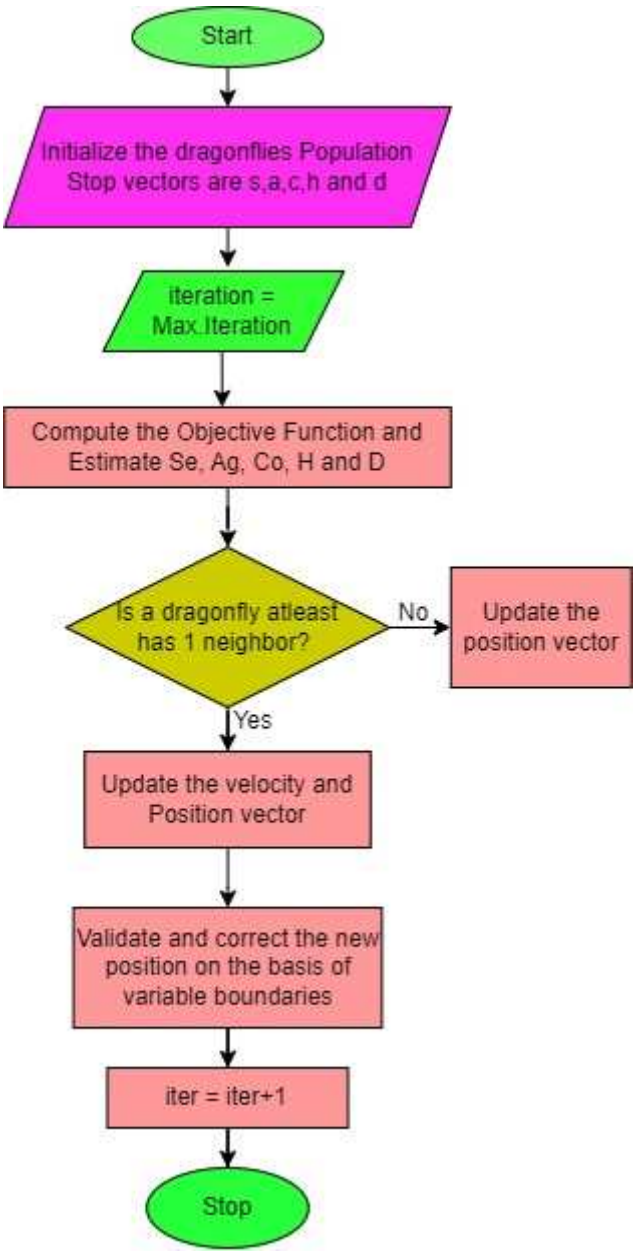

Figure 11. Flowchart of Dragonfly Optimization Algorithm.

The success of the feature selection methods outputs is analysed through the significance of the p-value from t-Test. Table 3 illustrates the p-value significance for the EHO and Dragon fly algorithm feature selection methods for the four DR techniques. As shown in the Table 3 that the EHO and Dragon fly algorithm Feature selection methods are not showing any significant p-values among the classes for all four DR methods. This p-value will be a precursor to quantify the presence of outliers, non-linear and non-Gaussian variables among the classes after feature selection methods.

Table 3. p-Value significant for Feature Selection method from t-test for different DR Techniques.

Feature selection	DR Techniques	Bessel Function	Discrete Cosine Transform (DCT)	Least Square Linear Regression (LSLR)	Artificial Algae Algorithm (AAA)

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 15 August 2023 doi:10.20944/preprints202308.1033.v1

	Class	DP	NDP	DP	NDP	DP	NDP	DP	NDP
EHO	P value <0.05	0.9721	0.9998	0.994	0.9996	0.9961	0.9999	0.9466	0.9605
Dragon Fly	P value <0.05	0.99985	0.876	0.9956	0.998	0.9951	0.99931	0.9936	0.9977

6. Classification Techniques

There are ten classification techniques used in this research such as Non-Linear Regression (NLR), Linear Regression (LR), Gaussian Mixture Model (GMM), Expectation Maximum (EM), Bayesian Linear Discriminant Classifier (BLDC), Logistic Regression (LoR), Softmax Discriminant Classifier (SDC), Support Vector Machine (SVM) with linear kernel, Support Vector Machine (SVM) with polynomial kernel, and Support Vector Machine (SVM) with radial basis function (RBF) kernel.

1. Non-Linear Regression (NLR)

The behavior of a system is expressed through mathematical equations to facilitate representation and analysis, ultimately aiming to determine an accurate best-fit line between classifier values. Non-linear regression introduces non-linear and random variables (a, b) to capture the complexity of the system. The primary objective of non-linear regression is to minimize 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.

Non-linear models require more attention due to their inherent complexity, and researchers have devised various methods to mitigate this complexity, such as the Levenberg-Marquardt and Gauss-Newton methods. Estimating parameters for non-linear systems is achieved through least square methods, aiming to minimize the residual sum of squares. Iterative techniques, including the Taylor series, steepest descent method, and Levenberg-Marquardt's method, Zhang et al., [32], can be employed for non-linear equations. The Levenberg-Marquardt technique is widely used for estimating non-linear least squares, offering advantages, and producing reliable results through an iterative process.

Authors assume a represented model,

$$z_i = f(x_i, \theta) + \varepsilon_i, \qquad where\ i = 1,2,3, \dots, n \tag{10}$$

Here, x_i *and* $z_i$are the independent and dependent variables of the ith iteration.
$\theta = (\theta_1, \theta_2, .., \theta_m)$ are the parameters and ε_i are the error terms that follows $N(0, \sigma^2)$.
The residual sum of squares is given by,

$$S_u(\theta) = \sum_{i=1}^{n} [z_i - f(x_i, \theta)]^2 \tag{11}$$

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

$$(H + \tau I)(\theta_0 - \theta_1) = g \tag{12}$$

Where, $g = \frac{\partial S_u(\theta)}{\partial \theta} \Big| \theta = \theta_o$ and $H = \frac{\partial^2 S_u(\theta)}{\partial \theta \partial \theta'} \Big| \theta = \theta_1$, τ is a multiplier and I is the identity matrix.

The goodness of fit of the model is assessed using the Mean Square Error (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, where N represents the overall number of experimental values

$$\text{MSE} = \frac{1}{N} \sum_{(i=1)}^{N} (y_i - y_i^{\wedge})^2 \tag{13}$$

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 15 August 2023 doi:10.20944/preprints202308.1033.v1

Steps to be followed for non-linear regression are, the initial parameters are initialized, and curves are generated 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.

2. Linear Regression (LR)

In the analysis of gene expression data, linear regression is a suitable method for obtaining the best-fit curve, as the expression levels in genes exhibit only minor variations. To identify the most informative genes, a feature selection process is performed by comparing the training dataset with 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 [33] 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.

$$g_\theta = \theta_0 + \theta_1 x \tag{14}$$

Where, $\theta_i$represents the parameters. The objective is to select the range of $\theta_o \; and \; \theta_1$that ensures g_θ closely approximates y in the training dataset (x, y).

$$R\,(\theta_0, \theta_1) = \frac{1}{2m}\sum_{i=1}^{m}(g_\theta(x^i) - y^i)^2 \tag{15}$$

where m represents the total number of samples in the training dataset. For a linear regression model with n variables, the hypothesis function becomes.

$$g_\theta = \theta_0 x_0 + \theta_1 x_1 + \cdots + \theta_n x_n \tag{16}$$

and the cost function is given by.

$$R\,(\theta) = \frac{1}{2m}\sum_{i=1}^{m}(g_\theta(x^i) - y^i)^2 \tag{17}$$

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

$$\frac{\delta}{\delta\theta_j}\, g(\theta) = \frac{\delta}{\delta\theta_j}\sum_{i=1}^{m}(g_\theta(x^i) - y^i)^2 \tag{18}$$

To update the parameter value θ_j, the following equation is used:

$$\theta_{j(new)} = \theta_{j(old)} + \beta\, \frac{1}{m}\sum_{i=1}^{m}(g_\theta(x^i - y^i)x_j^i) \tag{19}$$

where β represents the learning rate, and θ_j is continuously computed until convergence is reached. In this study, β is set to 0.01.

The algorithm for linear regression involves the following steps:

- Feature selection parameters, obtained from algorithms such as Bessel Function, DCT, LSLR and AAA, are used as input for 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 defined to minimize the squared error between the observed data and the predictions.
- The solutions are found by equating the derivatives of $\theta_0 \; and \; \theta_1$to zero.
- Steps 2, 3, and 4 are repeated to obtain coefficients that yield the minimum squared error.

3. Gaussian Mixture Model (GMM)

Gaussian Mixture Model (GMM) is a well-known unsupervised learning technique in machine learning used for pattern recognition and signal classification. It involves integrating related objects based on clustering techniques. By classifying similar data, GMM [34] facilitates the prediction and computation of unrated items within the same category. GMM encompasses both hard and soft clustering techniques and utilizes Gaussian mixture model distribution for data analysis. In GMM, the data is generated using Gaussian distribution techniques, where each GMM consists of multiple Gaussian distributions (referred to as "g"). The probability density function of GMM combines these distributed components linearly, enabling easier analysis of the generated data. For random value

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 15 August 2023 doi:10.20944/preprints202308.1033.v1

generation in a vector form "a" within an n-dimensional sample space χ, if "a" follows a Gaussian distribution, the probability distribution function is expressed as:

$$p(a) = \frac{1}{(2\pi)^{n/2}|\Sigma|^{1/2}} e^{-(\frac{1}{2})(a-\mu)^T\Sigma^{-1}(a-\mu)} \quad (20)$$

Here, μ represents the mean vector in the n-dimensional space, and Σ 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 [35], with each component having its individual vector specifications in the distribution curve. The mixture distribution equation is given by:

$$P_Q(a) = \sum_{i=1}^{k} \propto_j . p(a|\mu_j, \Sigma_j) \quad (21)$$

In this equation, $\propto_j$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.

4. Expectation Maximum (EM)

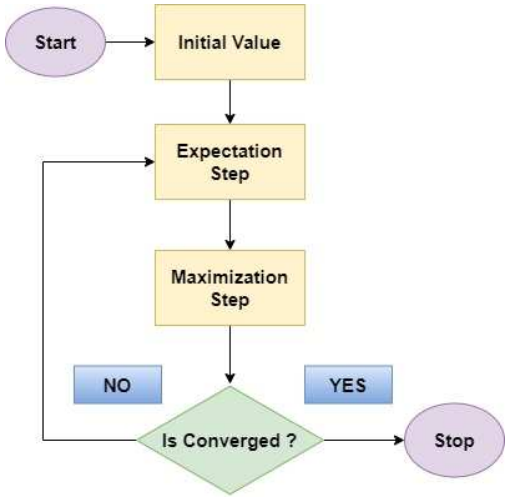

Figure 12. Flow diagram of Expectation Maximum.

The EM algorithm [36] 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 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.

$$L\,(\theta; X, Y) = p(X, Y|\theta) \quad (22)$$

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

The maximum likelihood estimation is obtained as:

$$L(\theta; X) = p\,(X|\theta) = \sum_Y p(X, Z|\theta) \quad (46)$$

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

$$Q(\theta|\theta\hat{}(t)) = Ez|x_i\; \theta\hat{}(t)\; [\log\; L\;(\theta; X, Y)] \quad (23)$$

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

$$\theta\hat{}(t+1) \;=\; \arg\max Q(\theta|\theta\hat{}(t)) \quad (48)$$

The Expectation and Maximization steps are iteratively repeated until a converged sequence of values is reached.

5. Bayesian Linear Discriminant Classifier (BLDC)

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 15 August 2023 doi:10.20944/preprints202308.1033.v1

The Bayesian Linear Discriminant Classifier (BLDC) [37] 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 of $\{B, m\}$ represents G. The matrix B corresponds to the training vector, and a denotes the filtered signal. β denotes the inverse variance of the noise, and C denotes the sample size. The prior distribution of x is expressed as

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

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 α is a hyperparameter obtained from data forecasting, while l represents the assigned vector number. The weight x follows a Gaussian distribution with zero mean and a small value contained in ε. By applying Bayes' rule, the posterior distribution of x can be computed as:

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

For the posterior distribution, the mean vector υ and the covariance matrix X should satisfy certain norms in the equation (50) and (52). The nature of the posterior distribution is predominantly Gaussian. Specifically:

$$\upsilon = \beta\, (\beta B B^T + H'(\alpha))^{-1} B a \tag{53}$$

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

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

$$p(\hat{a}|\beta, \alpha, \widehat{b, G}) = \int p(\hat{a}|\beta, \widehat{b, x}) p(x|\beta, \alpha, G) dy \tag{24}$$

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

6. Logistic Regression (LoR)

Logistic Regression (LoR) has proven to be effective in classifying diseases such as diabetes, cancer, and epilepsy. In this context, the author considers a function y that represents the disease status, ranging from 0 to 1 to indicate normal and diabetic patients respectively. Gene expressions are represented by a vector $x = x_1, x_2, .., x_m$. where each element x_j corresponds to the expression level of the jth gene. By constructing a model-based approach using 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:

$$Logit\,\{\Pi(x)\} = \upsilon_0 + \sum_{j=1}^{q} \upsilon_j x_j^* \tag{56}$$

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 15 August 2023 doi:10.20944/preprints202308.1033.v1

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

$$1(\upsilon_0, \upsilon) = \sum_{j=1}^{n}\{y_i \log(\pi_i) + (1-\pi_i)\} - \frac{1}{2\tau^2}\|\upsilon\|^2 \quad (25)$$

where τ is a parameter that limits the shrinkage of υ near 0, represents $\pi_i = \pi(x_i)$ as defined by the model [38][39], and $\|\upsilon\|^2$, denotes the Euclidean length of $\upsilon = \upsilon_1, \upsilon_2, \ldots, \upsilon_p$. The selection of q and τ are determined through the parametric bootstrap method, which imposes constraints on accurate error prediction. Initially, υ=0, for the purpose of computing 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 normal. Any probability below the threshold is considered indicative of normal patients, while values above the threshold indicate diabetic patients.

7. Softmax Discriminant Classifier (SDC)

The purpose of Softmax Discriminant Classifier (SDC) is to determine and identify the group to which a specific test sample belongs [40]. It involves weighing the distance between the training samples and the test sample within a particular class or group of data. The training set, denoted as Z,

$$Z = [Z_1, Z_2, .., Z_q] \in \Re^{c\times d} \quad (526)$$

consists of samples from distinct classes named q, $Z_q = [Z_1^q, Z_2^q, .., Z_{dq}^q] \in \Re^{c\times d_q}$. 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 $K \in \Re^{c\times 1}$, it 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:

$$h(K) = \arg\max Z_w^i \quad (27)$$

$$h(K) = \arg\max_i \log\left(\sum_{j=1}^{d_i} \exp(-\lambda \left\|v - v_j^i\right\|_2)\right) \quad (60)$$

In these equations, $h(K)$represents the distance between the i^{th} class and the test samples. $\lambda > 0$, serves as a penalty cost factor. If K is identified with the i^{th} class, then v and v_j^icorrespond to the same characteristic function, and $\left\|v - v_j^i\right\|_2$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 Support Vector Machine (SVM) classifier is a significant machine learning approach widely used for classification problems, particularly in non-linear regression-based multilayer perceptron [41]. In this study, three distinct methods are explored for data classification:

8. SVM - Linear: This method utilizes a linear kernel to classify the data.
9. SVM - Polynomial: This approach involves the use of a polynomial kernel for data classification.
10. SVM - Radial Basis Function (RBF): The RBF kernel is employed in this method for classifying 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 Support Vector Machine (SVM) depend on the chosen classifier and the data being utilized. 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.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 15 August 2023 doi:10.20944/preprints202308.1033.v1

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

The process of implementing 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 α_i,

$$Max\ L_D = \sum_{i=1}^{l} \alpha_i - \frac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{l} \alpha_i \alpha_j y_i y_j (X_i . X_j)\, v \quad (62)$$

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

Step 2: The next step involves solving the quadratic programming problem to obtain the optimal separating hyperplane. The data points with non-zero Lagrangian multipliers ($\propto_i > 0$) 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 centre points, which are representative points from the dataset. Each centre point acquires the points around them.

Step 5: If there are six centre points representing each cluster, the SVM training data is obtained using kernel methods.

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

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

The hyper plane and support vectors are used to separate linearly separable and nonlinearly separable data.

6.1. 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. [42] was followed to conduct the k-fold cross-validation process. Initially, the dataset was divided into k equally sized subsets or "folds". For each fold, denoted as i, the model was trained on the remaining data excluding the i^{th} fold, and then tested on the i^{th} fold. 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 k-fold cross-validation is its ability to provide a more reliable estimate of the model's performance compared to a simple train-test split, as it maximizes the utilization of available data. In our study, 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 by monitoring the Mean Square Error (MSE) proposed by Wang et al. [43], which is defined as follows:

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

Where O_j is the observed value at time j, T_j is the target value at model j;

j=1and 2, and N is the total number of observations per epoch in our case it is 2870. As the training progressed the MSE value reached 1.0 E-12 within 2000 iterations.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 15 August 2023 doi:10.20944/preprints202308.1033.v1

Table 4. Confusion matrix for Diabetic Detection.

Truth of Clinical Situation		Predicted Values	
		Diabetic	Normal
Actual Values	Diabetic	TP	FN
	Normal	FP	TN

In the case of lung cancer detection, the following terms can be defined as:

True Positive (TP): A patient is correctly identified as diabetic class.

True Negative (TN): A patient is correctly identified as a non-diabetic class.

False Positive (FP): A patient is incorrectly identified as diabetic class when they are in non-diabetic class.

False Negative (FN): A patient is incorrectly identified as non-diabetic class when they are in diabetic class.

Table 5 reveals the training and testing MSE performance of the classifiers without feature selection Method for four Dimensionality Reduction Techniques. The training MSE are always varied between 10^{-04} to 10^{-10}, while the testing MSE varies from 10^{-04} to 10^{-08}. SVM (RBF) classifier in AAA DR techniques without feature selection method settled at minimum training and Testing MSE of 1.93E-10 and 1.77E-08 respectively. The minimum testing MSE will be a marker of the better classifier performance. As shown in the Table 5 that higher the values of testing MSE leads to the poorer performance of the classifier irrespective of the Dimensionality reduction Techniques.

Table 5. Training and Testing MSE Analysis of Classifiers for Various Dimensionality Reduction Technique without Feature Selection Methods.

Classifiers	**Bessel Function**		**Discrete Cosine Transform (DCT)**		**Least Square Linear Regression (LSLR)**		**Artificial Algae Algorithm (AAA)**	
	Training MSE	Testing MSE	Training MSE	Testing MSE	Training MSE	Testing MSE	Training MSE	Testing MSE
NLR	0.0000023	0.000176	6.41E-06	2.48E-05	7.75E-06	5.12E-05	2.91E-07	1.6E-05
LR	2.41E-05	9.51E-05	7.52E-06	3.11E-05	2.18E-07	4.66E-05	3.67E-08	1.45E-05
GMM	0.0000021	0.000175	5.72E-07	6.8E-06	3.09E-07	1.11E-05	3.76E-06	5.33E-06
EM	**1.62E-07**	**9.87E-06**	2.71E-06	1.3E-05	9.87E-07	1.99E-05	8.97E-09	7.3E-06
BLDC	0.0000014	0.000253	2.86E-07	3.94E-05	4.74E-06	5.28E-05	1.43E-07	1.64E-05
LoR	0.0000012	0.000289	9.47E-06	3.58E-05	8.69E-06	4.54E-05	9.26E-08	1.45E-05
SDC	0.0000019	0.000203	3.66E-06	1.07E-05	2.47E-06	1.86E-05	2.31E-09	5E-06
SVM (L)	0.0000031	0.00027	8.92E-06	2.89E-05	1.09E-05	4.01E-05	4.13E-09	8.2E-06

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 15 August 2023 doi:10.20944/preprints202308.1033.v1

SVM (Poly)	0.0000036	0.000211	3.36E-07	2.11E-05	1.29E-06	2.85E-05	7.84E-09	4.69E-06
SVM (RBF)	4.16E-07	8.3E-05	**1.57E-08**	**2.41E-06**	**3.22E-08**	**5.64E-06**	**1.93E-10**	**1.77E-08**

Table 6 exhibits the training and testing MSE performance of the classifiers with EHO feature selection Method for four Dimensionality Reduction Techniques. The training MSE is varied between 10^{-5} to 10^{-10}, while the testing MSE varies from 10^{-5} to 10^{-8}. SVM (RBF) classifier in AAA DR method with PSO feature selection is achieved minimum training and Testing MSE of $1.99\times 10^{-10}$and $\mathbf{2.5}\times 10^{-8}$ respectively. Bessel function DR method indicates slightly lower training and testing MSE values for the classifiers when compared to other three DR techniques. All the classifiers slightly improved the performance in the Testing MSE when compared to without Feature selection methods. This will indicate the enhancement of the classifier performance irrespective of Dimensionality Reduction Techniques.

Table 6. Training and Testing MSE Analysis of Classifiers for Various Dimensionality Reduction Technique with EHO Feature Selection Methods.

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

Table 7 represents the training and testing MSE performance of the classifiers with Dragon fly algorithm-based feature selection Method for four Dimensionality Reduction Techniques. The

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 15 August 2023 doi:10.20944/preprints202308.1033.v1

training MSE are always varied between 10^{-6} to 10^{-9}, while the testing MSE varies from 10^{-5} to 10^{-8}. SVM (RBF) classifier with Dragon fly feature selection method settled at minimum training and Testing MSE of $1.66\times 10^{-9}$and $\mathbf{3.25}\times 10^{-8}$ respectively. In this feature selection method, all classifiers are enhanced in their training and testing performance. This will be indicated by the improvement of the accuracy, MCC and Kappa parameters of the classifier performance irrespective of the type of Dimensionality Reduction Techniques.

Table 7. Training and Testing MSE Analysis of Classifiers for Various Dimensionality Reduction Technique with Dragon fly Feature Selection Methods.

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

6.2. Selection of target

The target value for the non-diabetic case (T_{ND}) is taken at the lower side of zero to one (0→1) scale and this mapping is made according to the constraint of

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

whereμ_i is the mean value of input feature vectors for the N number of Non-diabetic Features taken for classification. Similarly, the target value for the Diabetic case (T_{Dia}) is taken at the upper side of zero to one (0→1) scale and this mapping is made based on

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

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 15 August 2023 doi:10.20944/preprints202308.1033.v1

whereμ_j is the average value of input feature vectors for the M number of Diabetic cases taken for classification. Note that the target value T_{Dia} would be greater than the average value of $\mu_i$and μ_j. The difference between the selected target values must be greater than or equal to 0.5, which is given by:

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

Based on the above constraints, the targets T_{ND} and T_{Dia} for Non-Diabetic and Diabetic patient output classes are chosen at 0.1 and 0.85 respectively. After selecting the target values, the Mean Squared Error (MSE) is used for evaluating the performance of machine learning Classifiers.

Table 8. Selection of Optimum 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 is EM steps. Criterion: MSE
EM	test point likelihood probability 0.13,cluster Probability of 0.45, with convergence rate of 0.631, Criterion: MSE
BDLC	P(y), Prior probability: 0.5, Class mean: 0.85,0.1; Criterion: MSE
Logistic regression	Threshold H$\boldsymbol{\theta}$(x) <0.48 with Criterion: MSE
SDC	γ=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 Criteria: MSE
SVM(Polynomial)	C: 0.76, Coefficient of the kernel function (gamma): 10, Class weights: 0.5, Convergence Criteria: MSE
SVM(RBF)	C: 1, Coefficient of the kernel function (gamma): 100, Class weights: 0.86, Convergence Criteria: MSE

1. Results and Discussion

The research uses standard ten-fold testing and training in which 10% of input features are employed for testing, whereas 90% are employed for training. The choice of performance measures is significant in evaluating classifier performance. The confusion matrix is used to evaluate the performance of Classifiers, especially in binary classification (i.e., classification into two classes, such as Diabetic or non-diabetic from the pancreas Micro array Genes). It can be used to calculate performance metrics such as Accuracy, F1 score, MCC, Error Rate, FM Metrics, and Kappa, commonly used to evaluate the model's overall performance. Table 9 depicts the parameters associated with the classifiers for performance Analysis.

Table 9. Performance Metrics.

Metrics	Formula	Assessment focus
Accuracy	$Acc = \frac{(TN + TP)}{(TN + FN + TP + FP)}$	Fraction of predictions that are correct

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 15 August 2023 doi:10.20944/preprints202308.1033.v1

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

Here are the definitions of TP, TN, FP, and FN:

True positive (TP): A prediction that was correct and the actual value was positive.

True negative (TN): A prediction that was correct and the actual value was negative.

False positive (FP): A prediction that was incorrect and the actual value was negative.

False negative (FN): A prediction that was incorrect and the actual value was positive.

The performance of the classifier was evaluated using a few metrics, including accuracy, F1 score, MCC, error rate, FM metric, 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. 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 positive instances and to correctly identify 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 the following tables.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 15 August 2023 doi:10.20944/preprints202308.1033.v1

Table 10 expresses the performance analysis of the ten classifiers based on parameters like Accuracy, F1 Score, MCC, Error Rate, F Measure, and Kappa values for the four Dimensionality Reduction method without two feature selection methods. It is noted from the Table 9 that EM Classifier in the Bessel function DR techniques is settled at middle accuracy of 61.42%, F1 Score of 54.23% with moderate Error rate of 38.57% and F Measure of 57.28%. The EM Classifier is also exhibiting a low value of MCC 0.3092 and Kappa value of 0.2645. The SVM (linear) Classifier in Bessel function DR Method displays low accuracy of 52.85% with high Error Rate of 47.15%, and F1 score of 40% along with F Measure of 41.57%. The MCC and Kappa for SVM (linear) classifier is placed at the lower ebb of 0.06324 and0.05714 respectively. All the classifiers are poorly performed across the performance metrics in the Bessel function DR Techniques. This is due to inherit property of Bessel function and it is displayed in the non-negative values of statistical parameters. The SVM (RBF) Classifier for DCT DR Technique is maintained at accuracy of 88.57%, with low Error Rate of 11.42% F1 Score of 81.81% and F Measure of 82.15%. The MCC and Kappa values of SVM (RBF) classifier is at 0.7423 and 0.7358 respectively. The AAA DR technique in SVM (RBF) classifier attains high accuracy of 90% with low error rate of 10% and F1 Score of 84.44% with F measure of 84.97%. The MCC and kappa values for SVM(RBF) classifier settled at 0.7825 and 0.772 respectively. Irrespective of the Dimensionality Reduction Techniques all the classifiers are maintained at accuracy within the range of 52%-85%. This is due the inherit limitation of the Dimensionality Reduction Techniques. Hence, it is suggested to incorporate the Feature selection methods towards the enhancement of classifier's performance.

Table 10. Average Performance of Classifiers with different DM techniques Without Feature Selection Methods.

Dimensionality Reduction	Classifiers	Parameters					
		Accuracy (%)	F1Score (%)	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

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 15 August 2023 doi:10.20944/preprints202308.1033.v1

	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 Square 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

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 15 August 2023 doi:10.20944/preprints202308.1033.v1

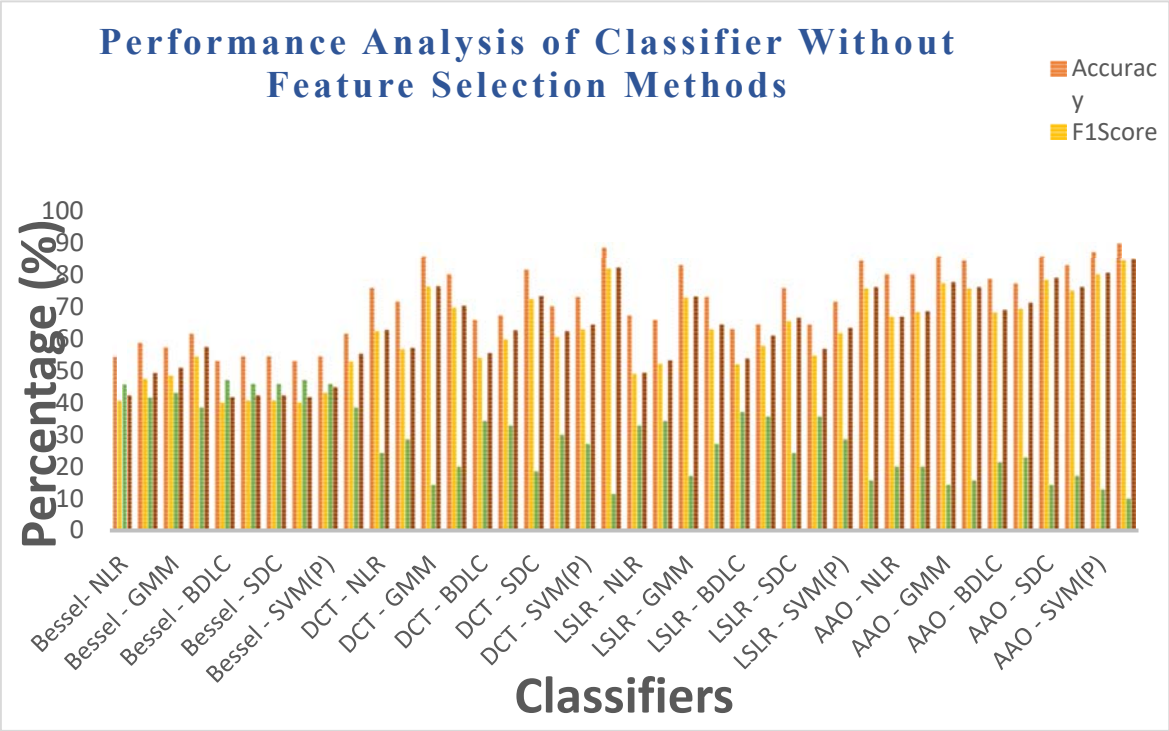

Figure 13. Performance Analysis of Classifiers without Feature Selection Methods.

Figure 13 depicts the performance analysis of the ten classifiers based on parameters such as Accuracy, F1 Score, Error Rate, and F measure values for the four Dimensionality Reduction method without feature selection methods. It is depicted from the Table 10 that EM Classifier in the Bessel function DR techniques is settled at middle accuracy of 61.42%, F1 Score of 54.23% with moderate Error rate of 38.57% and F Measure of 57.28%. The SVM (linear) Classifier in Bessel function DR Method displays low accuracy of 52.85% with high Error Rate of 47.15%, and F1 score of 40% along with F Measure of 41.57%. All the classifiers are poorly performed across the performance metrics in the Bessel function DR Techniques. The SVM (RBF) Classifier for DCT DR Technique is maintained at accuracy of 88.57%, with low Error Rate of 11.42% F1 Score of 81.81% and F Measure of 82.15%. The AAA DR technique in SVM (RBF) classifier attains high accuracy of 90% with low error rate of 10% and F1 Score of 84.44% with F measure of 84.97%.

Table 11 reveals the performance analysis of the ten classifiers for the four Dimensionality Reduction method with EHO feature selection method. It is examined from the Table 10 that SVM(RBF) Classifier in the AAA DR techniques is settled at high accuracy of 95.71%, F1 Score of 92.68% with low Error rate of 4.28% and F Measure of 92.71%. The SVM(RBF) Classifier is also displays a high value of MCC 0.897 and Kappa value of 0.8965. The SVM(Linear) Classifier for Bessel function DR Technique once again is placed in the lower ebb of accuracy of 50%, with high Error Rate of 50% F1 Score of 36.36% and F Measure of 37.79%. The null value of MCC and Kappa are attained by SVM (Linear) classifier which is a very peculiar phenomenon of the classifiers performance. All the classifiers exhibit enhanced accuracy in the DCT, LSLR and AAA DR techniques. The effect of EHO feature selection is not revealing any appreciable improvements in the classifiers for Bessel function DR Method.

Table 11. Average Performance of Classifiers with different DM techniques With Elephant Herding Optimization (EHO) Feature Selection Method.

Dimensionality Reduction	Classifiers	Parameters					
		Accuracy (%)	F1 Score (%)	MCC	Error Rate	FM (%)	Kappa

 doi:10.20944/preprints202308.1033.v1

					(%)		
	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
Bessel	**LoR**	54.2857	40.7407	0.0813	45.7142	42.1831	0.0743
Function	**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
	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
Discrete	**BLDC**	85.7142	77.2727	0.6757	14.2857	77.594	0.6698
Cosine	**LoR**	81.4285	71.1111	0.5845	18.5714	71.5542	0.5767
Transform	**SDC**	85.7142	77.2727	0.6757	14.2857	77.594	0.6698
(DCT)	**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
	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
Least Square	**BLDC**	75.7142	65.3061	0.4952	24.2857	66.4364	0.4757
Linear	**LoR**	72.8571	61.2244	0.4310	27.1428	62.2841	0.4140
Regression	**SDC**	80	68.1818	0.5424	20	68.4653	0.5377
(LSLR)	**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	**NLR**	81.4285	73.4693	0.6236	18.5714	74.7409	0.5991
Algorithm	**LR**	87.1428	80	0.7165	12.8571	80.4984	0.7069
(AAA)	**GMM**	85.7142	78.2608	0.6918	14.2857	78.9352	0.6788

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 15 August 2023 doi:10.20944/preprints202308.1033.v1

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 displays the performance analysis of the ten classifiers s for the four Dimensionality Reduction methods with EHO feature selection methods. It is also observed from the Table 11 that SVM(RBF) Classifier in the AAA DR techniques is settled at high accuracy of 95.71%, F1 Score of 92.68% with low Error rate of 4.29% and F Measure of 92.71%. The SVM (linear) Classifier for Bessel Function DR Technique is settled in the lower end of accuracy of 50%, with high Error Rate of 50% F1 Score of 36.36% and F Measure of 37.79%. All the classifiers are poorly performed in the Bessel function DR techniques.

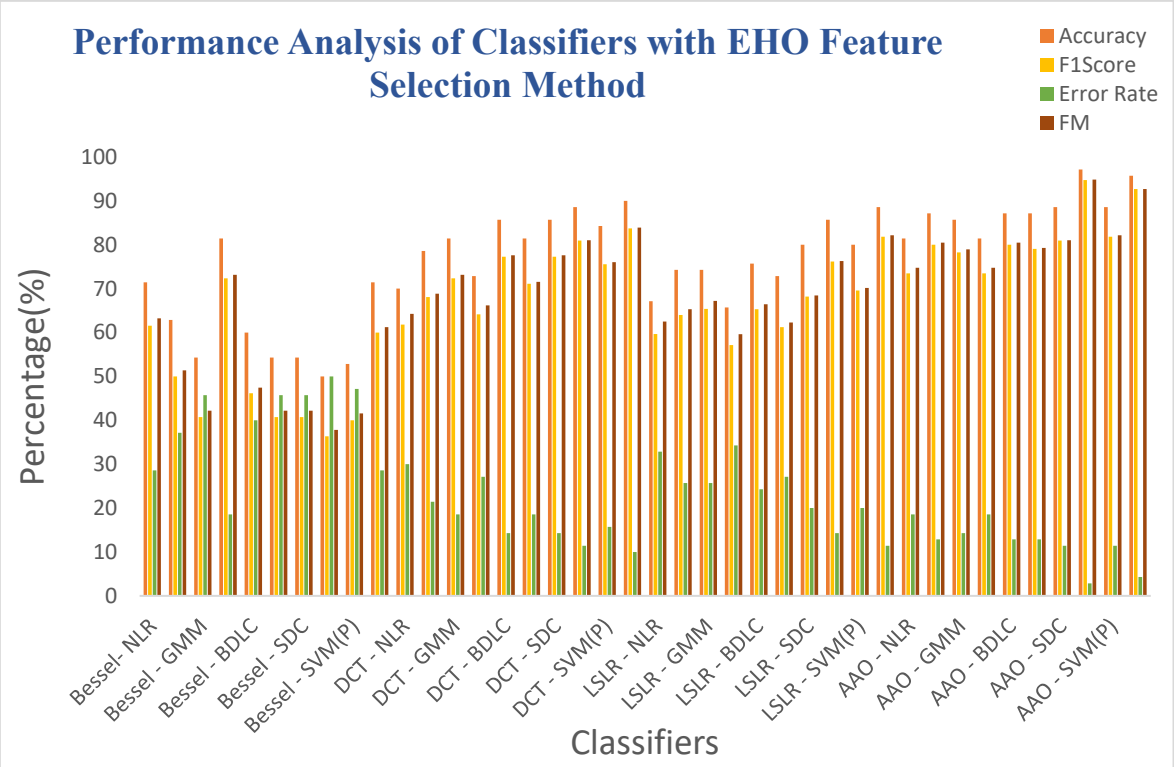

Figure 14. Performance Analysis of Classifiers with EHO Feature Selection Methods.

Table 12 explores the performance analysis of the ten classifiers for the four Dimensionality Reduction method with Dragon Fly feature selection method. It is mentioned from the Table 12 that SVM(RBF) Classifier in the AAA DR techniques is settled at high accuracy of 94.28%, F1 Score of 90.47% with low Error rate of 5.72% and F Measure of 90.57%. The SVM (RBF) Classifier is also exhibits a high value of MCC 0.866 and Kappa value of 0.864. The SVM (polynomial) Classifier for Bessel function DR Technique is assigned at the lower accuracy of 58.57%, with high Error Rate of 41.43% F1 Score of 43.13% and F Measure of 44.17%. The MCC and Kappa values of SVM (polynomial) classifier is at 0.1364and 0.1287 respectively. Except SVM (RBF) classifier all the other nine classifiers in the Bessel function DR method attains accuracy of less than 78%. SVM (RBF) classifier in DCT DR and LSLR DR Method achieved high accuracy of 91% and 90% respectively.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 15 August 2023 doi:10.20944/preprints202308.1033.v1

Table 12. Average Performance of Classifiers with different DM techniques With Dragon Fly Feature Selection Method.

Dimensionality Reduction	Classifiers	Parameters					
		Accuracy (%)	F1Score (%)	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 Square 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

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 15 August 2023 doi:10.20944/preprints202308.1033.v1

	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

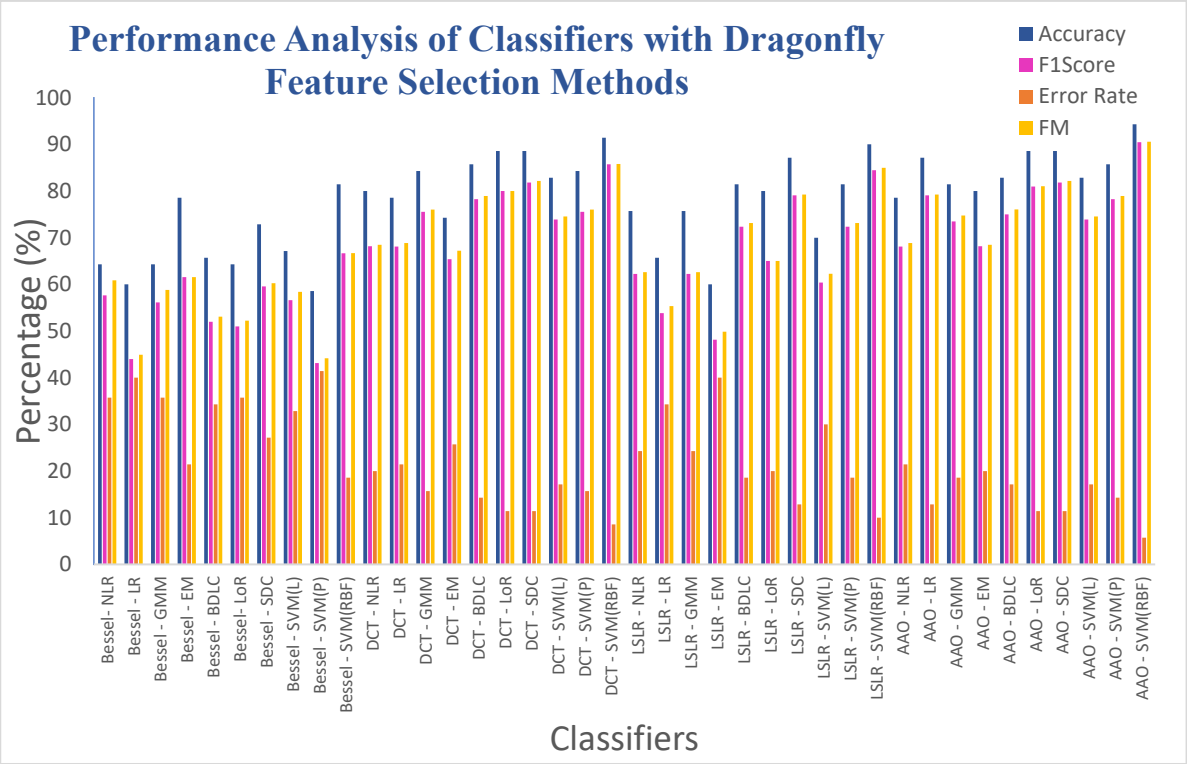

Figure 15. Performance Analysis of Classifiers with Dragon fly Feature Selection Methods.

Figure 15 demonstrates the performance analysis of the ten classifiers for the four Dimensionality Reduction methods with Dragon fly feature selection methods. It is also examined from the Table 12 that SVM(RBF) Classifier in the AAA DR techniques is settled at high accuracy of 94.28%, F1 Score of 90.47% with low Error rate of 5.72% and F Measure of 90.57%. The SVM (polynomial) Classifier for Bessel function DR Technique is placed at the lower accuracy of 58.57%, with high Error Rate of 41.43% F1 Score of 43.13% and F Measure of 44.17%. Across four DR methods SVM (RBF) classifier achieved individual accuracy of more than 81%. The classifiers performance in Bessel function DR method with Dragon fly Feature selection is settled at poor category only.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 15 August 2023 doi:10.20944/preprints202308.1033.v1

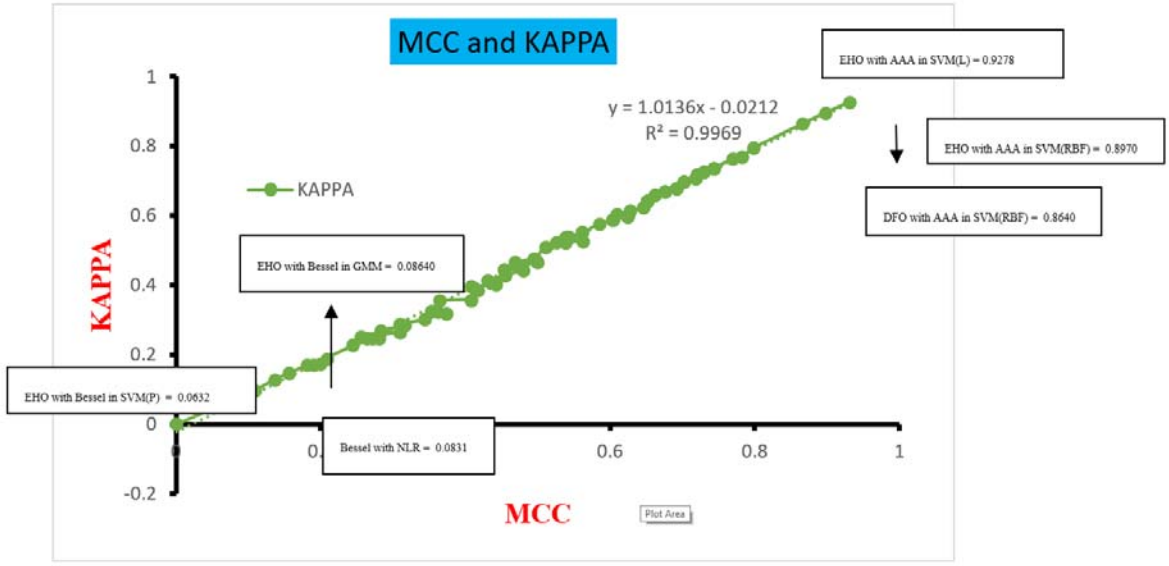

Figure 16. Performance of MCC and Kappa Parameters across the classifier for four DR Techniques without and with Two Feature Selection Methods.

Figure 16 displays the Performance of MCC and Kappa Parameters across the classifiers for four DR Techniques without and with Two Feature Selection Methods. The MCC and Kappa are the benchmark parameter which indicates the outcomes of the classifiers for different inputs. As in this research there are three categories of inputs like dimensionally reduced without Feature selection, with EHO and Dragon fly Feature selection methods. The classifiers performance is observed through the attained MCC and Kappa values for these inputs. The average MCC and Kappa values from the classifiers are at 0.2984 and 0.2849 respectively. A methodology is devised to identify the performance of the classifiers with reference to figure 14. The MCC values are divided into three ranges like 0.0 - 0.25, 0.251-0.54 and 0.55-0.9. The performance of the classifiers is poor in the range 1 and there is a steep increase in the MCC Vs Kappa slope in the region 2 of the MCC values. The region 3 of MCC is settled at higher performance of the classifiers without any glitches.

Figure 17 shows Histogram performance of Error Rate and MCC (%) Parameters across the classifiers for four DR Techniques without and with Two Feature Selection Methods. From the figure 17 it is also identified that maximum error rate is settled at 50% and Maximum MCC at 90%. The Histogram of Error rate is skewed at the right side of the graph which indicates that for any DR method and irrespective of Feature selection method the classifier's error rate does not go beyond 50%. The histogram of MCC across the classifier depicts the sparser in the edges and covers more points in the middle area.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 15 August 2023 doi:10.20944/preprints202308.1033.v1

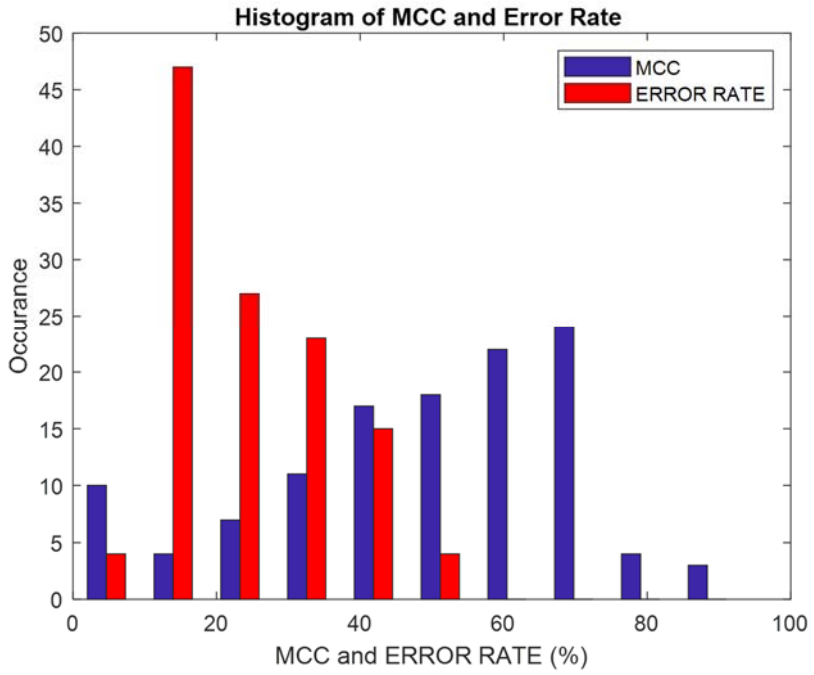

Figure 17. Performance of Error Rate and MCC (%) Parameters across the classifier for four DR Techniques without and with Two Feature Selection Methods.

6.3. Computational Complexity

The analysis of the classifiers in this study considers their computational complexity, which is determined based on the size of the input (denoted as O(n)). A lower computational complexity, indicated by O(1), is desirable as it indicates that the complexity remains constant regardless of the input size. However, as the number of inputs increases, the computational complexity will also increase. It is noteworthy that in this research, the computational complexity is independent of the input size, which is a favourable characteristic for any algorithm. If the computational complexity increases logarithmically with the increase in 'n', it is represented as O(logn). Additionally, all the classifiers used in this study are hybrid models that incorporate dimensionality reduction techniques and feature selection methods in their classification process.

Table 13 presents the Computational Complexity of the Classifiers for the four Dimensionality Reduction Techniques without incorporating Feature selection methods. A noteworthy observation from the table is that the computational complexity of all 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 computational complexity of O(n3 logn), while the Discrete Cosine transform, Least Square Linear Regression, and Artificial Algae Algorithm exhibit higher computational complexity with improved accuracy, represented by O(2n4log2n), O(2n5log4n), and O(2n5log8n) 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. Computational Complexity of the Classifiers for Different Dimensionality Reduction Method without Feature selection methods.

Classifiers	DR Method			
	Bessel Function	Discrete Cosine Transform (DST)	Least Square Linear Regression (LSLR)	Artificial Algae Algorithm (AAA)

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 15 August 2023 doi:10.20944/preprints202308.1033.v1

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 inferred the Computational Complexity of the Classifiers for the four Dimensionality Reduction Techniques when utilizing the EHO Feature selection method. The table reveals that the computational complexity of all the classifiers is relatively similar, while their performance demonstrates significant accuracy. Similar to the case without feature selection, the Expectation Maximum classifier exhibits higher computational complexity of O(n5 logn) along with remarkable accuracy. Regarding the DCT, LSLR, and AAA classifiers, they achieve similar computational complexity to the SVM(RBF) classifier with O(2n6 log2n), O(2n7 log4n), and O(2n7 log8n) 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. Computational Complexity of the Classifiers for Different Dimensionality Reduction Method with EHO Feature selection method.

Classifiers	**DR Method**			
	Bessel Function	**Discrete Cosine Transform (DCT)**	**Least Square 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	$O(2n^4 \log n)$	$O(2n^4 \log n)$	$O(2n^5 \log 2n)$	$O(2n^5 \log 4n)$

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 15 August 2023 doi:10.20944/preprints202308.1033.v1

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 Computational Complexity of the Classifiers for the four Dimensionality Reduction Techniques using the Dragonfly feature selection method. It is evident from the table that the computational complexity of all the classifiers is relatively similar, while their performance exhibits a significant level of accuracy. Notably, all four dimensionality reduction techniques demonstrate the highest computational complexity compared to their counterparts. Specifically, Bessel Function, DCT, LSLR, and AAA classifiers achieve a computational complexity of O(8n5log 2n), O(8n5log 2n), O(8n6log 4n), and O(8n6log 8n) respectively. Regarding accuracy, 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.

Table 15. Computational Complexity of the Classifiers for Different Dimensionality Reduction Method with Dragon Fly Feature selection method.

Classifiers	**DR Method**			
	Bessel Function	**Discrete Cosine Transform (DST)**	**Least Square 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	$O(4n^3 \log 2n)$	$O(4n^3 \log 2n)$	$O(4n4 \log 2n)$	$O(4n^4 \log 4n)$
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)	$O(8n^5 \log 2n)$	$O(8n^5 \log 2n)$	$O(8n^6 \log 4n)$	$O(8n^6 \log 8n)$

Comparison of previous works

As mentioned in the Table 16 it is notice that, majority of the machine learning classifiers such as SVM(RBF), Naïve Bayes, Logistic Regression, Decision tree, Non-linear regression, random forest, multilayer perceptron, and Deep neural networks are utilized to classify the diabetics, based on the clinical data base. All the classifiers accuracy is at the range of 67% - 95%. The current study is based on microarray gene to detect diabetes and SVM(RBF) achieved accuracy of 95%.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 15 August 2023 doi:10.20944/preprints202308.1033.v1

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) [44]	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) [45]	Diabetes: 12,000 age between 18–100 Age (Mean): 73	Training set - 66% ; tuning set 17% ; test set - 17%	RNN, CLST Memory (C-LSTM)	ACC - 79
3.	Hertroijs et al. (2018) [46]	Total: 105814 age(mean) : greater than 18	Training set (90%) test set (10%) 5-fold cross-validation	Latent Growth Mixture Modelling (LGMM)	ACC: 92.3
4.	ArellanoCampos et al. (2019) [47]	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) [48]	Total: 140 diabetes: 14 imbalanced age: 12–90	Training set (70%) test set (30%) fivefold cross-validation, holdout validation	BT, SVM (L)	ACC: 91
6.	Choi et al. (2019) [49]	Total: 8454 diabetes: 404 age: 40–72 PIDD	10-fold cross-validation	LoR, LDA, QDA, KNN	ACC: 78, 77 76, 77
7.	Akula et al. (2019) [50]	Practice Fusion Dataset total: 10,000 Age between : 18–80	Training set: 800; test set:10,000	KNN, SVM, DT, RF, GB, NN, NB	ACC: 86
8.	Xie et al. (2019) [51]	Total: 138,146 diabetes: 20,467 age: 30–80	Training set is around 67% and test set is around 33%	SVM, DT, LoR, RF, NN, NB	ACC: 81, 74, 81, 79, 82, 78
9.	Bernardini et al. (2020) [52]	Total: 252 diabetes: 252 age: 54–72	10-fold cross-validation	Multiple instance learning boosting	ACC: 83
10.	Zhang et al. (2020) [53]	Total: 36,652 age: 18–79	10-fold cross-validation	LoR, classification, and regression tree, GB, ANN, RF, SVM	ACC : 75, 80, 81, 74, 86, 76

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 15 August 2023 doi:10.20944/preprints202308.1033.v1

11.	Jain et al. (2020) [54]	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) [55]	Pima Indian data set	Hold out k-fold cross-validation	Stacking multi-layer perceptron, SVM, LoR	ACC: 78
13.	Haneef et al. (2021) [56]	Total 44,659 age 18–69 data is imbalanced	Training set (80%) test set (20%)	LDA	ACC: 67
14.	Deberneh et al. (2021) [57]	Total: 535,169 diabetes: 4.3% prediabetes: 36% age: 18–108	10-fold cross-validation	RF,SVM, XGBoost	ACC: 73, 73, 72
15.	Zhang et al. (2021) [58]	Total: 37,730 diabetes: 9.4% age: 50–70 Imbalanced	Training set is around 80% test set is around 20% 10-fold cross-validation	Bagging boosting, GBT, RF, GBM	ACC: 82
16.	This article	Nordic islet transplantation programme	10-fold 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; RRN – Recurrent Neural Network; CLST Memory – Convolutional Long Short Term Memory; BT – Bagged Tree; KNN – k-Nearest Neighbour; GB – Gradient Boost; NN – Neural Network; k-NN – k-Nearest Neighbour; GBT – Bagging Boost GBT; ACC – Accuracy;.

6.4. Limitation and Major Outcomes

The findings of this study may be limited to the specific population of diabetic mellitus type 2 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 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 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 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, better patient outcomes. Overall, this research contributes valuable insights to the field and lays the foundation for further investigations into the early detection and management of diabetic mellitus type 2 patients.

6.5. Conclusion

The results obtained from the BF-DR technique, the classifiers exhibited lower accuracy and other performance metrics, 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. Especially, 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 90%, 88.57%, and 95.71% for DCT, LSLR, and AAA respectively. With the use of Dragonfly feature selection method, which also showed promising results with the classifiers achieved high accuracy values of 81.42%, 91.42%, 90%, and 95.71% 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 across different dimensionality reduction techniques. In conclusion, this research article presents a novel method for detecting type II diabetes mellitus using microarray gene data. The future work will be in the direction of Convolution Neural Network (CNN), Deep Learning Networks (DNN), LSTM and Hyper parameter tuning of classifiers.

Supplementary Materials: Nil

Author Contributions: Conceptualization, C.D.; Methodology, C.D.; Software, C.D.; Validation, H.R.; Formal analysis, H.R.; Investigation, C.D.; Resources, C.D. and H.R.; Data curation, H.R.; Writing— original draft, C.D.; Writing—review and editing, H.R.; Visualization, C.D.; 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 data that support the findings of this study are available from the corresponding author upon reasonable request.

References

1. "Facts & Figures," International Diabetes Federation. [Online]. Available: https://idf.org/about-diabetes/facts-figures/
2. Pradeepa R, Mohan V. Epidemiology of type 2 diabetes in India. Indian J Ophthalmol. 2021 Nov;69(11):2932-2938. doi: 10.4103/ijo.IJO_1627_21. PMID: 34708726; PMCID: PMC8725109.
3. Chockalingam, Sriram, Maneesha Aluru, and Srinivas Aluru. "Microarray data processing techniques for genome-scale network inference from large public repositories." Microarrays 5.3 (2016): 23.
4. Herman, W. H., Ye, W., Griffin, S. J., Simmons, R. K., Davies, M. J., Khunti, K., & Wareham, N. J. (2015). 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, 38(8), 1449-1455.
5. Strianese, O., Rizzo, F., Ciccarelli, M., Galasso, G., D'Agostino, Y., Salvati, A., &Rusciano, M. R. (2020). Precision and personalized medicine: how genomic approach improves the management of cardiovascular and neurodegenerative disease. Genes, 11(7), 747.
6. Abul-Husn, N. S., & Kenny, E. E. (2019). Personalized medicine and the power of electronic health records. Cell, 177(1), 58-69.
7. Schnell, O., Crocker, J. B., & Weng, J. (2017). Impact of HbA1c testing at point of care on diabetes management. Journal of diabetes science and technology, 11(3), 611-617.
8. Lu, H., Chen, J., Yan, K., Jin, Q., Xue, Y., & Gao, Z. (2017). A hybrid feature selection algorithm for gene expression data classification. Neurocomputing, 256, 56-62.
9. American Diabetes Association Professional Practice Committee; 2. Classification and Diagnosis of Diabetes: Standards of Medical Care in Diabetes—2022. Diabetes Care 1 January 2022; 45 (Supplement_1): S17–S38. https://doi.org/10.2337/dc22-S002

10. Jakka A, Jakka VR. Performance evaluation of machine learning models for diabetes prediction. International Journal Innovation Technology Explore Eng Regular Issue. 2019; 8 (11):1976–80. https://doi.org/10.35940/ijitee.K2155.0981119
11. Radja M, Emanuel AWR. Performance evaluation of supervised machine learning algorithms using different data set sizes for diabetes prediction. In: 2019 5th international conference on science in information technology (ICSITech). 2019. https://doi.org/10.1109/icsitech46713.2019.8987479
12. Dinh A, Miertschin S, Young A, Mohanty SD. A data-driven approach to predicting diabetes and cardiovascular disease with machine learning BMC Med Inform Decis Mak. 2019;19(1):1–15. https://doi.org/10.1186/s12911-019-0918-5
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(6):e15431. https://doi.org/10.2196/15431.
14. Muhammad LJ, Algehyne EA, Usman SS. Predictive supervised machine learning models for diabetes mellitus. SN Comput Sci. 2020;1(5):1–10. https://doi.org/10.1007/s42979-020-00250-8
15. Kim H, Lim DH, 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(11):5597. https://doi.org/10.3390/ijerph18115597
16. Lawi, A., & Syarif, S. (2019, October). Performance evaluation of naive Bayes and support vector machine in type 2 diabetes Mellitus gene expression microarray data. In Journal of Physics: Conference Series (Vol. 1341, No. 4, p. 042018). IOP Publishing.
17. [17]Ciaramella, A., &Staiano, A. (2019). On the role of clustering and visualization techniques in gene microarray data. Algorithms, 12(6), 123.
18. Velliangiri, S., &Alagumuthukrishnan, S. J. P. C. S. (2019). A review of dimensionality reduction techniques for efficient computation. Procedia Computer Science, 165, 104-111.
19. Parand, K., &Nikarya, M. (2019). New numerical method based on generalized Bessel function to solve nonlinear Abel fractional differential equation of the first kind. *Nonlinear Engineering*, *8*(1), 438-448.
20. Bell, W. W. (1967). Special Functions For Scientists And Engineers, Published simultaneously in Canada by D.
21. Kalaiyarasi, M., &Rajaguru, H. (2022). Performance Analysis of Ovarian Cancer Detection and Classification for Microarray Gene Data. BioMed Research International, 2022.
22. N. Ahmed, T. Natarajan, and K. R. Rao,"Discrete cosine transform," IEEE Transactions on Computers, vol. C-23, no. 1, pp. 90–93, 1974.
23. J. Epps and E. Ambikairajah, Use of the discrete cosine transform for gene expression data analysis, 2004.
24. Hotelling, H. (1933). Analysis of a complex of statistical variables into principal components. Journal of Educational Psychology, 24(6), 417–441. https://doi.org/10.1037/h0071325
25. Hastie, Trevor, et al. The elements of statistical learning: data mining, inference, and prediction. Vol. 2. New York: springer, 2009.
26. Uymaz, S. A., Tezel, G., &Yel, E. (2015). Artificial algae algorithm (AAA) for nonlinear global optimization. Applied soft computing, 31, 153-171.
27. Prabhakar, S. K., & Lee, S. W. (2020). An integrated approach for ovarian cancer classification with the application of stochastic optimization. IEEE access, 8, 127866-127882.
28. Wang, G. G., Deb, S., & Coelho, L. D. S. (2015, December). Elephant herding optimization. In 2015 3rd international symposium on computational and business intelligence (ISCBI) (pp. 1-5). IEEE.
29. Bharanidharan, N., &Rajaguru, H. (2021). Dementia MRI image classification using transformation technique based on elephant herding optimization with Randomized Adam method for updating the hyper-parameters. International Journal of Imaging Systems and Technology, 31(3), 1221-1245.
30. Mirjalili, S. (2016). Dragonfly algorithm: a new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems. Neural computing and applications, 27, 1053-1073.
31. Bharanidharan N, Rajaguru H. Performance enhancement of swarm intelligence techniques in dementia classification using dragonfly-based hybrid algorithms. Int J Imaging Syst Technol. 2019;1–18. https://doi.org/10. 1002/ima.22365
32. Zhang, Guanglu, Douglas Allaire, and Jonathan Cagan. "Reducing the Search Space for Global Minimum: A Focused Regions Identification Method for Least Squares Parameter Estimation in Nonlinear Models." Journal of Computing and Information Science in Engineering 23.2 (2023): 021006.
33. Draper, Norman R., and Harry Smith. Applied regression analysis. Vol. 326. John Wiley & Sons, 1998.
34. O. Llaha and A. Rista, "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), May 2021, pp. 94-102.

35. Prabhakar, Sunil Kumar, HarikumarRajaguru, and Seong-Whan Lee. "A comprehensive analysis of alcoholic EEG signals with detrend fluctuation analysis and post classifiers." *2019 7th International Winter Conference on Brain-Computer Interface (BCI)*. IEEE, 2019.
36. Liu, S., Zhang, X., Xu, L., & Ding, F. (2022). Expectation–maximization algorithm for bilinear systems by using the Rauch–Tung–Striebel smoother. *Automatica*, *142*, 110365.
37. Zhou, Weidong, et al. "Epileptic seizure detection using lacunarity and Bayesian linear discriminant analysis in intracranial EEG." IEEE Transactions on Biomedical Engineering 60.12 (2013): 3375-3381.
38. Hamid, Imad Yagoub. "Prediction of Type 2 Diabetes through Risk Factors using Binary Logistic Regression." Journal of Al-Qadisiyah for computer science and mathematics 12.3 (2020): Page-1.
39. Adiwijaya K, Wisesty UN, Lisnawati E, Aditsania A, Kusumo DS. Dimensionality reduction using principal component analysis for cancer detection based on microarray data classification. J Comput Sci. 2018; 14:1521-1530. https://doi.org/ 10.3844/jcssp.2018.1521.1530
40. F Zang, JS Zhang, "Softmax Discriminant Classifier", 3rd International Conference on Multimedia Information Networking and Security, pp. 16-20, 2011
41. Yao, X. J., et al. "Comparative classification study of toxicity mechanisms using support vector machines and radial basis function neural networks." *Analytica Chimica Acta* 535.1-2 (2005): 259-273.
42. Fushiki, Tadayoshi. "Estimation of prediction error by using K-fold cross-validation." Statistics and Computing 21 (2011): 137-146
43. Wang, Z., & Bovik, A. C. (2009). Mean squared error: Love it or leave it? A new look at signal fidelity measures. *IEEE signal processing magazine*, *26*(1), 98-117.
44. Maniruzzaman M, Kumar N, Abedin MM, Islam MS, Suri HS, El-Baz AS, Suri JS. Comparative approaches for classification of diabetes mellitus data: machine learning paradigm. Comput Methods Programs Biomed. 2017;152:23–34. https://doi.org/10.1016/j.cmpb.2017.09.004
45. Pham T, Tran T, Phung D, Venkatesh S. Predicting healthcare trajectories from medical records: a deep learning approach. J Biomed Inform. 2017;69:218–29. https://doi.org/10.1016/j.jbi.2017.04.00
46. Hertroijs DFL, Elissen AMJ, Brouwers MCGJ, Schaper NC, Köhler S, Popa MC, Asteriadis S, Hendriks SH, Bilo HJ, 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(3):681–8. https://doi.org/10.1111/dom.13148.
47. Arellano-Campos O, Gómez-Velasco DV, Bello-Chavolla OY, Cruz-Bautista I, Melgarejo-Hernandez MA, Muñoz-Hernandez L, Guillén LE, Garduño-Garcia JDJ, 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(1):1–10. https://doi.org/10.1186/s12902-019-0361-8
48. Deo R, Panigrahi S. Performance assessment of machine learning based models for diabetes prediction. In: 2019 IEEE healthcare innovations and point of care technologies, (HI-POCT). 2019. https://doi.org/10.1109/hi-poct45284.2019.8962811
49. Choi BG, Rha S-W, Kim SW, Kang JH, Park JY, 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(2):191. https://doi.org/10.3349/ymj.2019.60.2.191
50. Akula R, Nguyen N, Garibay I. Supervised machine learning based ensemble model for accurate prediction of type 2 diabetes. In: 2019 Southeast Con. 2019. https://doi.org/10.1109/southeastcon42311.2019.9020358
51. Xie Z, Nikolayeva O, Luo J, Li D. Building risk prediction models for type 2 diabetes using machine learning techniques. Prev Chronic Dis. 2019. https://doi.org/10.5888/pcd16.190109.
52. 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. ArtifIntell Med. 2020;105:101847. https://doi.org/10.1016/j.artmed.2020.101847
53. 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(1):1–10. https://doi.org/10.1038/s41598-020-61123-x
54. Jain S. A supervised model for diabetes divination. BiosciBiotechnol Res Commun. 2020;13(14, SI):315–8. https://doi.org/10.21786/bbrc/13.14/7
55. Kalagotla SK, Gangashetty SV, Giridhar K. A novel stacking technique for prediction of diabetes. Comput Biol Med. 2021;135:104554. https://doi.org/10.1016/j.compbiomed.2021.104554
56. Haneef R, Fuentes S, Fosse-Edorh S, Hrzic R, Kab S, Cosson E, Gallay A.Use of artifcial 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. https://doi.org/10.21203/rs.3.rs-139421/v1.
57. Deberneh HM, Kim I. Prediction of Type 2 diabetes based on machine learning algorithm. Int J Environ Res Public Health. 2021;18(6):3317. https://doi.org/10.3390/ijerph1806331

58. 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(10):4005–16. https://doi.org/10.1109/JBHI.2021.307711