

Article

Not peer-reviewed version

Machine Learning Meets Meta-heuristics: Bald Eagle Search Optimization and Red Deer Optimization for Feature Selection in Type II Diabetes Diagnosis

[Harikumar Rajaguru](#)^{*} and [Dinesh Chellappan](#)

Posted Date: 27 June 2024

doi: 10.20944/preprints202406.1948.v1

Keywords: Classifiers; Bald Eagle Search Optimization; Red Deer Optimization; Diabetic Detection; Performance Analysis; Feature extraction.



Preprints.org is a free multidiscipline platform providing preprint service that is dedicated to making early versions of research outputs permanently available and citable. Preprints posted at Preprints.org appear in Web of Science, Crossref, Google Scholar, Scilit, Europe PMC.

Copyright: This is an open access article distributed under the Creative Commons Attribution License which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Article

Machine Learning Meets Meta-heuristics: Bald Eagle Search Optimization and Red Deer Optimization for Feature Selection in Type II Diabetes Diagnosis

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.

* Correspondence: harikumarrajaguru@gmail.com

Abstract: This article investigates the effectiveness of feature extraction and selection techniques for enhancing the performance of classifiers accuracy in Type II Diabetes Mellitus (DM) detection using microarray gene data. To address the inherent high dimensionality of the data by employing three Feature Extraction (FE) methods are used namely, Short-Time Fourier Transform (STFT), Ridge Regression (RR), and Pearson Correlation Coefficient (PCC). To further refine the data, meta-heuristic algorithms like Bald Eagle Search Optimization (BESO) and Red Deer Optimization (RDO) are utilized for feature selection. The performance of seven classification techniques such as Non-Linear Regression - NLR, Linear Regression - LR, Gaussian Mixture Model - GMM, Expectation Maximization - EM, Logistic Regression - LoR, Softmax Discriminant Classifier - SDC, and Support Vector Machine with Radial Basis Function kernel - SVM-RBF are evaluated with and without feature selection. The analysis reveals that the combination of PCC with SVM-RBF achieved a promising accuracy of 92.85% even without feature selection. Notably, employing BESO with PCC and SVM-RBF maintained this high accuracy. However, the highest overall accuracy of 97.14% was achieved when RDO was used for feature selection alongside PCC and SVM-RBF. These findings highlight the potential of feature extraction and selection techniques, particularly RDO with PCC, in improving the accuracy of DM detection using microarray gene data.

Keywords: Classifiers; Bald Eagle Search Optimization; Red Deer Optimization; Diabetic Detection; Performance Analysis; Feature extraction.

1. Introduction

India faces a significant public health challenge with the escalating prevalence of diabetes. According to the International Diabetes Federation (IDF) Atlas data from 2021, a staggering 77 million adults in India already suffer from this chronic condition. Even more alarming are projections that this number will reach a critical mass of 134 million by 2045. This translates to roughly one in eight adults in India potentially having diabetes [1]. Particularly worrisome is the increasing number of young adults (below 40 years old) being diagnosed with type 2 diabetes.

Several factors contribute to this alarming rise. India's rapid urbanization is linked to lifestyle changes that heighten diabetes risk. These changes include reduced physical activity levels, increased consumption of processed foods, and rising obesity rates. Additionally, genetic predisposition, stress, and environmental factors all play a role [2]. Unfortunately, many diabetes cases remain undiagnosed until complications arise, highlighting the need for increased awareness and early detection strategies.

The human cost of diabetes in India is immense, with this disease ranking as the seventh leading cause of death. The economic burden is equally concerning. It's estimated that diabetes drains the Indian economy of approximately \$100 billion annually. Clearly, addressing this growing public

health crisis requires a multi-pronged approach that focuses on preventive measures for type 2 diabetes, early diagnosis for all types, and proper management of the condition [3].

1.1. Review of Related Works

Early and accurate diagnosis of diabetes is critical for effective management, but traditional methods like blood glucose tests have limitations. Microarray gene technology offers a promising alternative by analyzing gene expression patterns in the pancreas, potentially revealing early signs of the disease [4]. Recent advancements in machine learning algorithms have further enhanced their effectiveness for diabetes detection. A novel Convolutional Neural Network (CNN) architecture specifically designed for diabetes classification using electronic health records (EHR) data [5]. Their model achieved an impressive accuracy of 85.2%, demonstrating the potential of deep learning for diabetes diagnosis from readily available clinical data. The use of Explainable Artificial Intelligence (XAI) techniques with machine learning models for diabetes prediction [6]. Their XAI approach not only achieved high accuracy but also provided insights into the factors most influencing model predictions. This interpretability is crucial for clinicians to understand and trust the model's recommendations [7]. Certain investigations were made to use of machine learning models that combine various data sources, including gene expression data, clinical information, and lifestyle factors. Their findings suggest that integrating multimodal data sets can lead to more accurate and comprehensive diabetes risk prediction models [8]. Ensemble learning approaches continue to show strong performance. Specific algorithms like Random Forest and Deep Neural Networks have proven effective in various studies[9]. Microarray gene data analysis remains a valuable avenue for exploration. While traditional datasets have dominated research in machine learning for diabetes detection, microarray gene data offers a relatively unexplored avenue[10]. This presents an opportunity to investigate the effectiveness of various performance metrics such as accuracy, F1 Score, Mathew Correlation Coefficient, Jaccard Metrics, Error Rate and Kappa.

2. Materials and Methods

As depicted in Figure 1, this research utilizes a multi-stage approach for accurate diabetic detection. The first stage focuses on extracting relevant features from the microarray gene data. Three techniques are utilized for this purpose such as STFT, RR, PCC. The second stage, which is two roots, one is extracted data is directly given to classifiers to measure the performance and second one is given to feature selection process. In this research, FS used two meta-heuristic algorithms, BESO and RDO. These algorithms can help identify the most informative features from the initial extraction stage, potentially leading to a more efficient classification process. Finally, the third stage utilizes various classification techniques to categorize the data into diabetic and non-diabetic classes by using seven methods as NLR, LR, GMM, EM, LoR, SDC, and SVM-RBF.

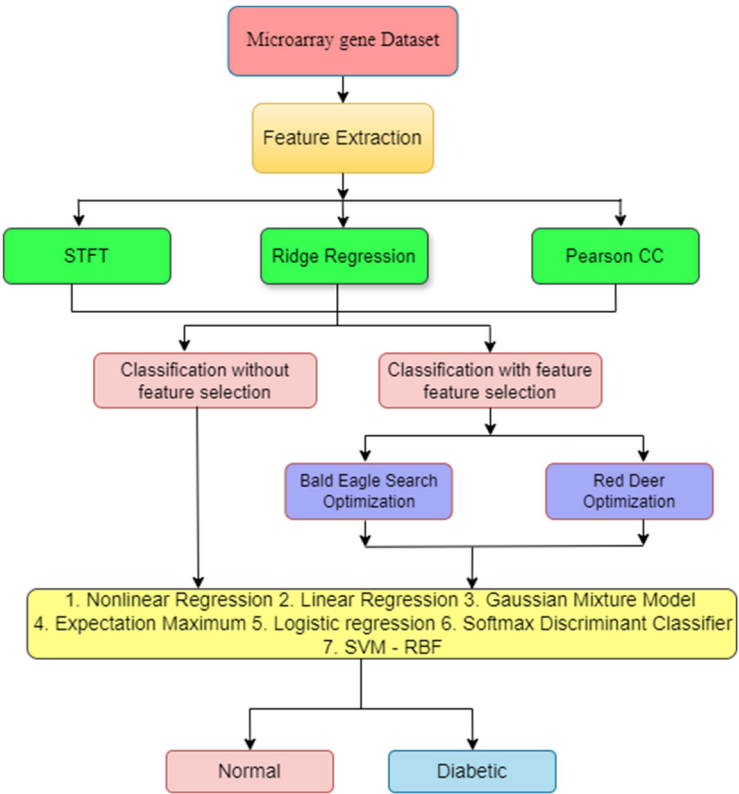


Figure 1. Overall flow diagram.

2.1. Dataset Details

Our research focused on utilizing microarray gene data from human pancreatic islets to detect diabetes and explore potential features associated with the disease. The data originated from the Nordic islet transplantation program (<https://www.ncbi.nlm.nih.gov/bioproject/PRJNA178122>) and included gene expression profiles from 57 healthy individuals and 20 diabetic patients. To ensure data quality and facilitate analysis, preprocessing steps were implemented. First, genes with the highest peak intensity per patient were selected, resulting in a dataset of 22,960 genes. Next, a logarithmic transformation (base 10) was applied to standardize the data across samples, ensuring a mean of 0 and a variance of 1. This step helps to normalize gene expression values and account for potential variations between samples. A key challenge in microarray data analysis is dealing with high dimensionality, referring to the large number of genes present. To address this, we employed feature extraction techniques. These techniques aim to reduce the number of features while retaining the most informative content relevant to diabetes detection [11]. Following feature extraction, feature selection techniques were applied to further refine the dataset and potentially improve classification accuracy. Specifically, two optimization algorithms were utilized, BESO [12] and RDO [13]. These algorithms helped identify the most relevant features for diabetic detection, further reducing the dimensionality of the data. Finally, to evaluate the performance and accuracy of diabetes classification, seven different classification algorithms were used. These algorithms will be discussed in detail later in the article.

2.2. Need for Feature Extraction (FE)

Microarray data analysis presents a challenge due to its high dimensionality. This means the data includes a vast number of genes, which can be overwhelming for classification algorithms. A high number of features can lead to a phenomenon called the “curse of dimensionality.” This occurs when the number of features [14] grows exponentially, making it difficult for classification algorithms to learn effective decision boundaries and potentially leading to overfitting. Feature extraction

techniques help address this challenge by reducing the number of features while retaining the most informative ones for diabetes detection. This process essentially focuses the analysis on the most relevant aspects of the gene expression data that contribute to differentiating between diabetic and non-diabetic samples. FE acts as a filter, selecting the most informative features from the vast amount of gene expression data. This allows the classification algorithms to work with a more manageable dataset and ultimately leads to more accurate diabetes detection.

2.2.1. Short-Time Fourier Transform (STFT)

The STFT offers a valuable technique for analyzing microarray gene expression data. It excels at capturing frequency domain information within specific time windows. This allows researchers to condense the dataset by extracting key features that represent how gene expression levels change over time and across various frequencies. Like its application in QRS complex detection by [15], STFT provides a time-frequency representation of gene expression data. This visualization facilitates exploration of how gene expression levels fluctuate across time and different frequency components. A key strength of STFT lies in its ability to pinpoint specific time intervals where frequencies are dominant. This allows researchers to identify genes with temporal patterns that activate during distinct time periods within the data. Additionally, STFT contributes to dimensionality reduction by extracting significant genes or groups of genes associated with specific frequency components [16]. This targeted approach streamlines the exploration of gene expression dynamics, ultimately aiding in the discovery of biologically relevant patterns within the data.

$$X[m, w] = \sum_{n=-\infty}^{\infty} x[n]w[n - m]e^{-iwn} \quad (1)$$

Where $x[n]$ is the input data where length N represents from $0, 1, 2, \dots, N-1$; $w[m]$ is the representation for STFT window with ranges from $0, 1, 2, \dots, M-1$. Moreover $i = \sqrt{-1}$ is the complex variable.

2.2.2. Ridge Regression (RR)

The analysis begins by focusing on smaller groups of samples. For each group, a feature matrix (X_i) is created. This matrix includes information about each sample as data point in a row, along with its corresponding outcome in an outcome vector (Y_i) [17]. Ridge regression is then applied to this local data to identify patterns. The estimates obtained by applying ridge regression to these local groups are referred to as local ridge regression estimates.

$$\hat{\beta}_i = (X_i^T X_i + \lambda_i I_p)^{-1} X_i^T Y_i \quad (2)$$

$$\hat{\beta}_{dist} = \sum_{i=1}^q (\omega_i \hat{\beta}_i) \quad (3)$$

Analyzing the estimation error in distributed ridge regression benefits from studying it within the framework of finite sample analysis for linear models. In this context, we consider the standard linear model, where Y represents the outcome variable, X is the feature matrix, b is the coefficient vector, and ε represents the error term. The outcome for each of the 'n' independent samples is represented by $Y \in \mathbb{R}^n$ a continuous vector with 'n' dimensions. The study utilizes a data matrix, X , to represent the features of the samples. This matrix has dimensions $n \times p$, where 'n' represents the number of samples and 'p' represents the number of features measured for each sample. We're also interested in a vector of coefficients, denoted by β . $\beta = (\beta_1, \beta_2, \beta_3, \dots, \beta_p)^T \in \mathbb{R}^n$. This vector has 'p' dimensions and contains unknown values, which aim to estimate through analysis. These coefficients will ultimately tell how much each feature contributes to the outcome variable we're trying to predict. This technique is employed for two main purposes: predicting the outcome variable for new samples and accurately estimating the influence of each feature (coefficient) on that outcome. However, it's important to consider that random errors or noise can significantly impact the outcome vector, $\varepsilon = (\varepsilon_1, \varepsilon_2, \dots, \varepsilon_n)^T \in \mathbb{R}^n$ which represents the actual values that are trying to predict [18].

In linear models, we often assume that the errors in the outcome variable (represented by the vector ε) are independent of each other, have an average value of zero, and a constant variance σ^2 . Ridge regression is a popular technique for estimating the coefficients (β). Here the equation for the same is,

$$\hat{\beta}(\lambda) = (X^T X + n\lambda I_p)^{-1} X^T Y \quad (4)$$

The ridge regression technique incorporates a parameter known as lambda (λ), which plays a crucial role in tuning the model. This parameter offers several advantages when it comes to estimating the coefficients (β). One key benefit is that it helps to 'shrink' the coefficients obtained from ordinary least squares regression, often leading to improved estimation accuracy. Now, let's consider a scenario where our data samples are spread across multiple locations or machines, represented by 'q' in total. In this case, we can partition the data for analysis as

$$X = \begin{bmatrix} X_1 \\ \vdots \\ X_q \end{bmatrix} \text{ and } Y = \begin{bmatrix} Y_1 \\ \vdots \\ Y_q \end{bmatrix}; \quad (5)$$

In distributed computing environments, where data is spread across multiple machines, approximate solutions for ridge regression are often preferred. This approach involves performing ridge regression locally on smaller subsets of the data (X_i, Y_i) at each machine[19]. One popular technique focuses on a one-shot weighting approach, where the final coefficients are obtained by combining the locally estimated coefficients using weighted sums. This weighting method offers a valuable starting point for more iterative techniques used in distributed regression. Additionally, this approach allows researchers to explore new characteristics of these one-shot weights.

To achieve this, local ridge regression with a regularization parameter λ_i is performed on each data subset (X_i, Y_i). The formula for these local ridge estimators is as follows:

$$\hat{\beta}_i(\lambda_i) = (X_i^T X_i + n_i \lambda_i I_p)^{-1} X_i^T Y_i \quad (6)$$

Here,

- $\hat{\beta}_i$ represents the locally estimated coefficient vector for a specific data subset.
- X_i denotes the design matrix for the i^{th} data subset.
- Y_i represents the outcome vector for the i^{th} data subset.
- λ_i is the regularization parameter for the local ridge regression on the i -th subset.
- I is the identity matrix with the same dimension as $X_i^T X_i$.

By using a weighted one-shot distributed estimation summation, the local ridge estimators from each data subset are combined as:

$$\hat{\beta}_{distributed} = \sum_{i=1}^q (\omega_i \hat{\beta}_i) \quad (7)$$

where,

- $\hat{\beta}_{distributed}$ represents the final combined coefficient vector obtained from all data subsets.
- ω_i represents the weight assigned to the local estimator from the i^{th} data subset.
- $\hat{\beta}_i$ represents the locally estimated coefficient vector for the i^{th} data subset (as defined earlier).

Unlike Ordinary Least Squares (OLS), the local ridge regression estimators we defined previously have some inherent bias. Due to this bias, there is no effect of imposing constraints on the weights used to combine them. This approach is particularly well-suited for data matrices (X) with any covariance structure (Σ). Assuming we have 'n' samples that are equally distributed across different machines, we can compute a local ridge estimator ($\hat{\beta}_1$) for each data subset[20]. Additionally, we can estimate local values for the signal-to-noise ratio (SNR, represented by $\hat{\sigma}_i^2$) and the noise level $\hat{\sigma}_i^2$ for each subset. To find the optimal weights for combining these local estimators, we consider three key factors: m, m0, and λ .

2.2.3. Pearson Correlation Coefficient (PCC)

Pearson's Correlation Coefficient (PCC) [21] serves as a powerful statistical tool for analyzing microarray gene expression data in diabetes detection. It measures the strength and direction of the linear relationship between gene expression levels. High PCC values, positive or negative, indicate that gene expression levels tend to fluctuate together. This co-fluctuation can reveal underlying biological processes. Genes with strong positive PCC may be regulated by similar mechanisms, potentially impacting diabetes development. Conversely, genes with strong negative PCC might play

opposing roles in cellular processes, with one potentially protective against diabetes while the other linked to disease progression.

By analyzing PCC values between gene pairs, we can identify potentially valuable genes for diabetes detection. Genes with strong positive correlations may act together to influence the diabetic state. Conversely, genes with strong negative correlations might represent opposing pathways with implications for disease development [22].

PCC analysis helps us understand how gene expression levels co-vary, ultimately aiding in the selection of informative gene sets that can be used to build robust models for accurate diabetes classification.

$$\rho(x, y) = \frac{\text{cov}(x, y)}{\sigma_x \sigma_y} \quad (8)$$

$$\text{cov}(x, y) = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{n-1} \quad (9)$$

$\text{cov}(x, y)$ Indicates the covariances between x and y .

$$\sigma_x = \sqrt{\frac{\sum_{i=1}^n (x_i - \bar{x})^2}{n-1}} \quad (10)$$

$$\sigma_y = \sqrt{\frac{\sum_{i=1}^n (y_i - \bar{y})^2}{n-1}} \quad (11)$$

σ_x, σ_y indicated the Standard Deviation of x and y .

From the Table 1, analyzing various features extracted from the data, STFT and RR shows minimal class differences in mean of 40.7 and variance is around 11,700 between diabetic and non-diabetic groups, suggesting they might not be strong discriminators for diabetes. Slight variations in Skewness and Kurtosis for most features hint at some distributional differences between the classes. Sample and Shannon Entropy values are identical within each class but differ between them, potentially indicating distinct underlying data patterns. Higuchi Fractal Dimension [23], however, stands out with a clear distinction (around 1.1 for STFT and 2.0 for Diabetic Class data) between classes for all features, suggesting its potential value in diabetes detection. Finally, Canonical Correlation Analysis (CCA) reveals a significant difference between diabetic and non-diabetic classes for the Diabetic Class data (0.4031 vs. 0.0675). This highlights CCA's potential for identifying features within the Diabetic Class data that are relevant for diabetes classification[24]. Overall, the table emphasizes the importance of analyzing various statistical parameters to understand the characteristics of different features and their potential role in differentiating diabetic and non-diabetic samples. The observed variations in skewness and kurtosis suggest the data may not follow a typical Gaussian (normal) distribution and might exhibit non-linear relationships. This can be further confirmed by visualizing the data using techniques like histograms, normal probability plots, and scatter plots of the feature extraction outputs.

Table 1. Statistical Analysis for Different Feature Extraction Techniques.

Statistical Parameters	STFT		Ridge Regression		Pearson CC	
	Dia P	Non-Dia P	Dia P	Non-Dia P	Dia P	Non-Dia P
Mean	40.7681	40.7863	0.0033	0.0025	0.0047	0.0045
Variance	11745.67	11789.27	1.3511	1.3746	0.0004	0.0004
Skewness	19.2455	19.2461	0.0284	-0.0032	0.0038	-0.0317
Kurtosis	388.5211	388.5372	0.6909	0.9046	-0.1658	-0.0884
Sample Entropy	11.0014	11.0014	11.4868	11.4868	11.4868	11.4868
Shannon Entropy	0	0	3.9818	3.9684	2.8979	2.9848
Higuchi Fractal Dimension	1.1097	1.1104	2.007	2.0093	1.9834	1.9659
CCA	0.4031		0.0675		0.0908	

Figure 2 shows the Normal Probability plot of STFT for both diabetic and non-diabetic class, from the data points 1 to 4 as reference points, and from 5 to 8 as upper bounds and from 9 to 12 as clustered variable points.

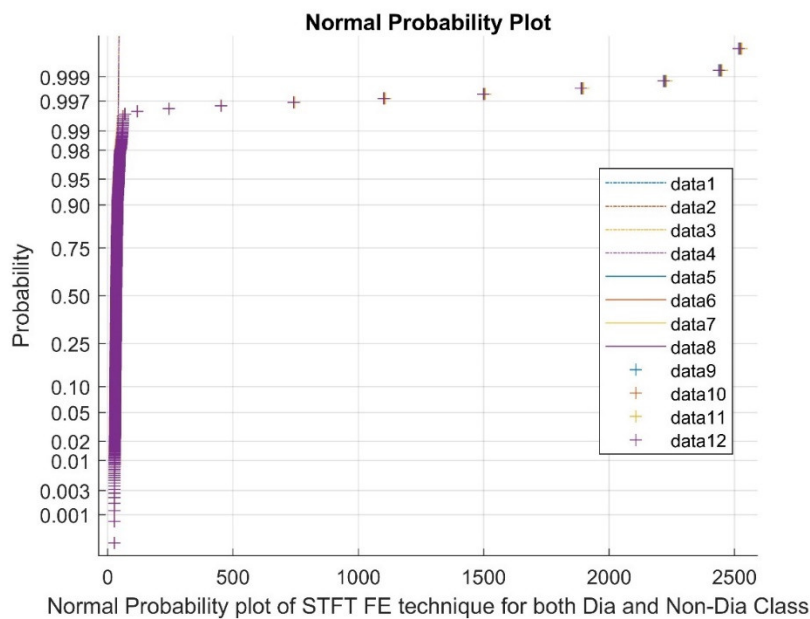


Figure 2. Normal plot of STFT FE techniques for both diabetic and non-diabetic class.

Figure 3 shows the histograms of the RR FE technique applied to both diabetic and non-diabetic classes. While the distributions appear somewhat bell-shaped, suggesting a possible tendency towards normality, a closer look reveals some deviations. Notably, there is an overlap between the distributions of the diabetic and non-diabetic classes. This overlap, along with potential deviations within each class distribution, suggests a more complex, non-linear relationship between the classes.

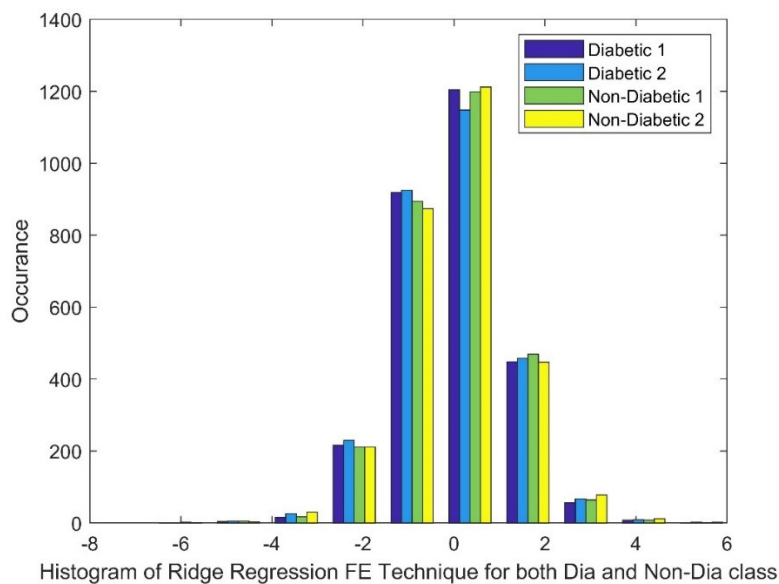


Figure 3. Histogram of RR FE techniques for both diabetic and non-diabetic class.

Figure 4 displays histogram of the PCC FE technique applied to both diabetic and non-diabetic classes. The markers on the x-axis represent patients, with $x(:,1)$ indicating patient 1 and $x(:,10)$ indicating patient 10. The histograms reveal a skewed distribution of values, suggesting a non-

normal data pattern. Additionally, a gap and potential non-linearity are observed in the data distribution for this method.

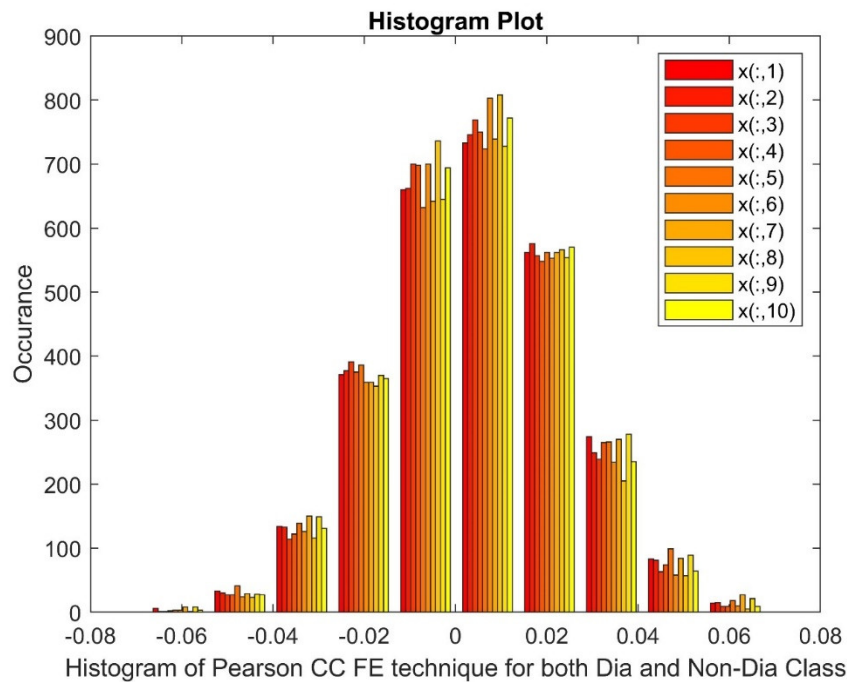


Figure 4. Histogram of PCC FE techniques for both diabetic and non-diabetic class.

From all the techniques of FE, it is revealed that, non-gaussian, non-linearity and techniques highlight the complexity of the data is observed across the classes and the potential for further feature selection is essential.

3. Feature Selection Method

As mentioned in section 1, there are 2 meta-heuristic algorithms namely BESO and RDO used as feature selection techniques that should be compared with and without FS method from FE techniques to classifiers for enhancement of classifier performance.

3.1. Bald Eagle Search Optimization (BESO)

Bald Eagle Search Optimization (BESO) is a recent addition to the field of Meta-heuristic Algorithms [25]. It draws inspiration from the hunting behavior of bald eagles, specifically their adeptness in locating and capturing prey. During the optimization process, BESO utilizes various search strategies modeled on these predatory behaviors. These strategies allow the search agents to explore the search space effectively, ultimately leading to the identification of optimal solutions for complex real-world problems.

Step 1: Simulating Exploration: Selecting the Search Area

The BESO algorithm mimics the exploration phase of a bald eagle's hunt through its first stage. In this stage, the algorithm determines a suitable search area for potential solutions, like how a bald eagle might scout a vast region for prey, which is mathematically stated as

$$P_{new}^t = P_{Best}^t + ar(P_{mean}^t - P_i^t) \quad (12)$$

A control parameter (a) set between 1.5 and 2 guides the search area's size, while a random number (r) between 0 and 1 introduces an element of chance, preventing stagnation in local optima. This exploration is further influenced by two key factors. The algorithm considers the best solution found so far (P_{Best}^t) to guide its search towards promising regions. Additionally, the average position of the current eagle population (P_{mean}^t) is factored in to ensure exploration extends beyond the immediate vicinity of the best solution and delves into new areas. By combining these elements -

a controlled search area size, random exploration, and consideration of both the best-found solution and the average population position – [26] BESO effectively navigates the search space during the first stage, laying the groundwork for identifying optimal solutions in later stages.

Step 2: Intensifying the Search

The second stage of BESO intensifies the search for optimal solutions within the exploration area identified in phase one. This mimics the behavior of a bald eagle that has located a promising hunting ground. Here, the search agents meticulously scan the defined region. They follow a spiral-like pattern, like how a bald eagle might meticulously search a lake for fish. This focused search pattern allows the algorithm to efficiently explore the selected area and identify potential solutions with greater precision. The mathematical expression as,

$$P_{new}^t = P_i^t + y_i X (P_i^t - P_{i+1}^t) + x_i X (P_i^t - P_{mean}^t) \quad (13)$$

Where,

$$\begin{aligned} x_i &= \frac{xr_i}{\max|xr|}, y_i = \frac{yr_i}{\max|yr|} \\ xr_i &= r_i X \sin(\theta_i), yr_i = r_i X \cos(\theta_i) \\ \theta_i &= a X \pi X rand, a \in (5,10) \\ r_i &= \theta_i + R + rand, R \in [0.5,2] \end{aligned}$$

The two random parameters, denoted as 'a' and 'R', control the search pattern within the chosen exploration area. These parameters influence the number of spirals undertaken by the search agents and the variation in their coiling shape. This element of randomness helps prevent the algorithm from getting stuck in suboptimal solutions. By introducing variation in the search pattern, BESO encourages exploration of diverse positions within the search area, ultimately leading to the identification of more accurate solutions.

Step 3: Convergence

The final stage of BESO, represents the convergence towards the optimal solution [27]. In this stage, the search agents gravitate towards the most promising position identified so far. This convergence process can be mathematically written as

$$P_{new}^t = rand X P_{Best}^t + x1_i X (P_i^t - C_1 X P_{mean}^t) + y1_i X (P_i^t - C_2 X P_{Best}^t) \quad (14)$$

Where,

$$\begin{aligned} x1_i &= \frac{xr_i}{\max|xr|}, y1_i = \frac{yr_i}{\max|yr|} \\ xr_i &= r_i X \sinh(\theta_i), yr_i = r_i X \cosh(\theta_i) \\ r_i &= \theta_i, \theta_i = a X \pi X rand, a \in (5,10) \end{aligned}$$

Two random parameters (C_1, C_2) play a crucial role in this stage. These parameters, ranging between 1 and 2, control the intensity of the search agents' movement towards the best solution. The algorithm utilizes a polar equation to mathematically model the "swooping" motion of the eagle. This mathematical framework ensures an efficient convergence process. Additionally, the mean position of the search agents can be factored in to introduce more diversity and intensity during the convergence phase, further guiding them towards the optimal solution.

3.2. Red Deer Optimization (RDO)

Red Deer Optimization (RDO) is a recent metaheuristic algorithm inspired by the fascinating mating behavior of red deer during the rut, or breeding season. Introduced in 2016 [28], RDO mimics this natural process to find optimal solutions to complex problems. The algorithm starts with a random population of individuals, representing potential solutions to the diabetic detection problem.

These individuals are analogous to red deer. The strongest individuals, like dominant males, are identified. These “stags” compete for influence over the remaining population, like how stags fight for harems of females during the rut. The outcome of this competition determines how solutions are combined to create new offspring, representing the next generation of potential solutions. Stronger “stags” have a greater chance of mating with more “hinds,” leading to a wider exploration of the solution space. Weaker individuals may still contribute, but to a lesser extent. This process balances exploration and exploitation [29] – key aspects of optimization. Through this iterative process of selection, competition, and reproduction, RDO progressively refines the population towards improved solutions for diabetic detection. The final population contains the most promising solutions identified by the algorithm[30].

RDO algorithm.

a) Initialization : Each RD is described by a set of variables, analogous to its genes. The number of variables (XNvar) corresponds to the number of genes. The values of these variables represent the potential contribution of each gene to diabetic detection. For instance, the set XNvar to 50 which means to account for investigations.

Initializing Individual Positions:

The position of each RD (X_i) is defined by this set of variables. Here, X_i would be a vector containing the values for each of the 50 genes (e.g., $[01, 02, \dots, 050]$). These initial values are assigned randomly.

b) Roaring: In the “roar stage,” each RD representing a potential gene selection for diabetic detection, can explore its surrounding solution space based on the microarray gene data. Imagine each RD has neighboring solutions in this multidimensional space. Here, a RD can adjust its position which means gene selection within this local area. The algorithm evaluates the fitness of both the original and the adjusted positions using a fitness function reflecting how well the gene selection differentiates between diabetic and non-diabetic cases.

An equation is used to update the RD's position,

$$Male_{new} = \begin{cases} male_{new} + a_1 \times (((UB - LB) \times a_2) + LB), & \text{if } a_3 \geq 0.5 \\ male_{new} - a_1 \times (((UB - LB) \times a_2) + LB), & \text{if } a_3 < 0.5 \end{cases} \quad (15)$$

Here, a_1, a_2, a_3 are the random factors between 0 and 1. $Male_{old}$ and $Male_{new}$ are the old and new positions of RD's. This process is akin to the RD exploring its surroundings and “roaring” if it finds a better location with a higher fitness value (B^*) [31]. Such successful adjustments elevate the RD to commander status, signifying a promising solution for diabetic detection. Conversely, if the exploration yields a position (A^*) with a lower fitness value compared to the original, the Red Deer remains a “stag.” These stags still have the chance to contribute in later stages of the algorithm. This roar stage essentially refines the initial population by focusing on RD's gene selections that demonstrate better potential for differentiating diabetic and non-diabetic cases based on the gene data.

c) Roar

Following the roar stage, commanders engage in a unique form of competition with the remaining stags. This competition doesn't involve literal combat, but rather an exchange of information about their gene selections.

$$New_1 = \frac{(Com + Stag)}{2} + b_1 \times (((UB - LB) \times b_2) + LB) \quad (16)$$

$$New_2 = \frac{(Com + Stag)}{2} - b_1 \times (((UB - LB) \times b_2) + LB) \quad (17)$$

Each commander interacts with a set of randomly chosen stags. The fight's outcome determines how much information, or details about their gene selections, the commander shares with the stags. Commanders with demonstrably better gene selections will share more information with the stags they compete with. Imagine commanders with promising solutions guiding the stags towards improvement. Here, commanders with demonstrably superior gene selections retain their position as leaders and share more knowledge with the stags. Conversely, if a stag's gene selection shows

potential during the competition, it might surpass the commander’s current selection. In such cases, the stag’s improved solution becomes the new leader, and the information flow is reversed.

d) Creation phase of harems:

Following the roar and fight stages, commanders are rewarded based on their performance. This reward system is reflected in the formation of harems. A harem consists of a commander and a group of hinds as female deer. The number of hinds a commander attracts is directly proportional to his success in the previous stages. Commanders with demonstrably superior gene selections will attract larger harems. Imagine a commander’s “power” being determined by how well its gene selection differentiates between diabetic and non-diabetic cases.

$$N. \text{ harem}_n = \text{round} \{p_n \cdot N_{hind}\}$$

(18)

Commanders with greater power will attract more hinds to their harems. Stags, on the other hand, are not included in harems. This reward system incentivizes the continued exploration and refinement of promising gene selections for diabetic detection.

e) The mating phase in RDO occurs in three key scenarios: 1. Commander Mating Within Harems - Each commander gets the opportunity to mate with a specific proportion (α) of the hinds within its harem. This mating metaphorically represents the creation of new gene selections based on combinations of the commander’s strong selection and those of the hinds. 2. Commander Expansion Beyond Harems - commanders can potentially mate with hinds from other harems. A random harem is chosen, and its commander gets the chance to “mate” with a certain percentage (β) which lies between 0 and 1) of the hinds in other harems. 3. Stag Mating - Stags also have a chance to contribute. They can mate with the closest hind, regardless of harem boundaries. This allows even less successful gene selections to potentially contribute to the next generation, introducing some level of diversity. By incorporating these three mating scenarios, RDO explores a combination of promising gene selections, ventures beyond established solutions, and maintains some level of diversity through stag mating.

f) Mating phase – New solutions are created. This process combines the strengths of existing gene selections from commanders, hinds, and even stags, promoting a balance between inheritance and exploration. This approach helps refine the population towards more effective gene selections for diabetic detection using your microarray gene data.

g) Building the Next Generation - RDO employs a two-pronged approach to select individuals for the next generation. A portion of the strongest RD is automatically selected for the next generation. These individuals represent the most promising gene selections identified so far. Additional members for the next generation are chosen from the remaining hinds and the newly generated offspring. This selection process often utilizes techniques like fitness tournaments or roulette wheels, which favor individuals with better fitness values.

h) RDO’s stopping criterion to determine the number of Iterations - a set number of iterations can be predetermined as the stopping point. The algorithm might stop if it identifies a solution that surpasses a certain threshold of quality for differentiating diabetic and non-diabetic segregation, a time limit might also be set as the stopping criterion. The parameters of each values involves in this algorithm as described in the following Table 2.

Table 2. Parameters of RDO.

S.No.	Parameters	Values	S.No.	Parameters	Values
1.	Initial Population (I)	100	6.	Beta (β)	0.4
2.	Maximum time of simulation	10 (s)	7.	Gamma (γ)	0.7
3.	Number of males (M)	15	8.	Roar	0.25
4.	Number of hinds (H)	I – M	9.	Fight	0.4
5.	Alpha (α)	0.85	10.	Mating	0.77

Table 3. Utilizing p-Values for Feature Selection in Diabetes Detection: A Comparison of various FE techniques.

Feature selection	DR Techniques	STFT		Ridge Regression		Pearson CC	
	Class	Dia P	Non-Dia P	Dia P	Non-Dia P	Dia P	Non-Dia P
BESO	P value <0.05	0.4673	0.3545	0.2962	0.2599	0.3373	0.3178
RDO	P value <0.05	0.4996	0.4999	0.4999	0.4883	0.4999	0.4999

Based on the p-value selection criteria employed in Table 2, features extracted using STFT, RR, and PCC might not be the most informative for diabetes detection. This conclusion is drawn from analyzing p-values obtained through t-tests for various FE methods for two FS techniques as BESO and RDO were revealed that p-values can serve as initial indicators to quantify the potential presence of outliers, non-linearity, and non-Gaussian data distributions within the classes. Further analysis may be necessary to identify the most statistically significant features for accurate diabetes classification.

3.3. Analyzing the Impact of Feature Extraction Methods Using Statistical Measures

Microarray data, with gene expression information, can be overwhelming due to its high dimensionality. Feature extraction techniques help streamline this data. Subsequently, the Pearson Correlation Matrix (PCM) comes into play for in-depth analysis. PCM calculates pairwise correlations between genes, essentially measuring of how gene expression levels varies from low to high change together across different samples. Values closer to 1 (positive correlation) indicate genes that tend to be expressed together, while values closer to -1 (negative correlation) suggest genes with opposite expression patterns[32]. A value around 0 signifies no significant correlation. By identifying clusters of genes with strong positive correlations (potentially sharing similar functions), PCM paves the way for discovering promising diabetic biomarkers. Figure 5 shows weak correlation between the classes. STFT BESO values for diabetic patients range from -0.69 to 0.98, while for non-diabetic patients they range from -0.94 to 0.78. Similarly, Figure 6 also indicates weak correlation between the classes is observed based on RR BESO values. These values range from -0.54 to 0.86 for diabetic patients and -0.69 to 0.68 for non-diabetic patients.

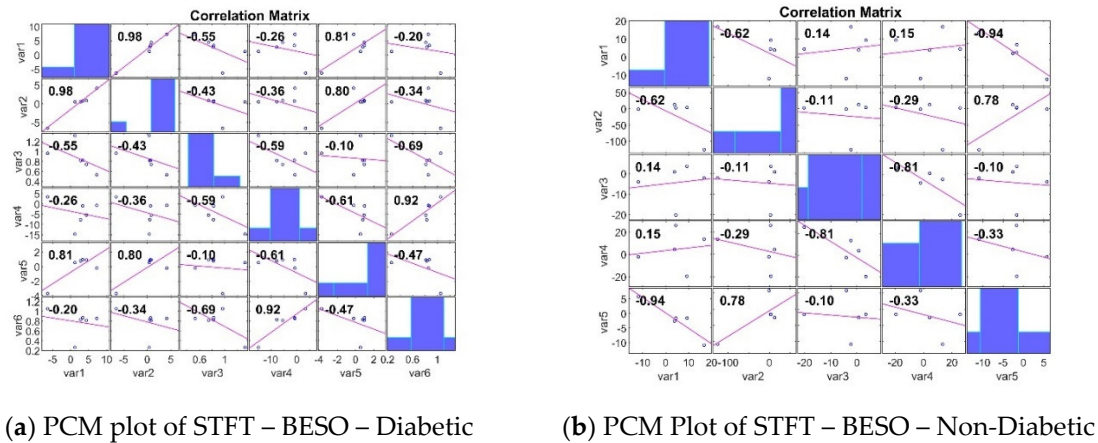
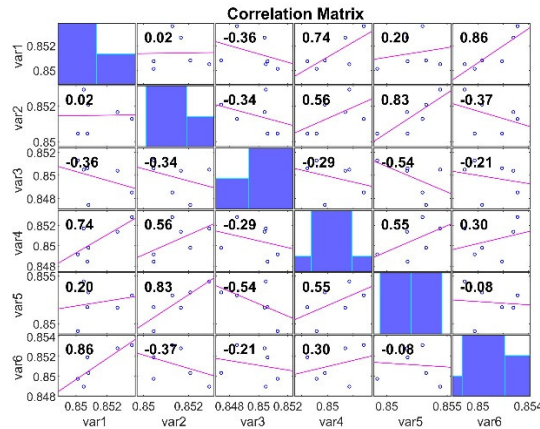
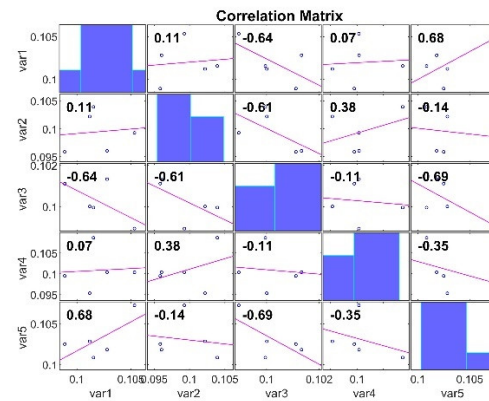


Figure 5. PCM Plots for STFT FE Method.



(c) PCM plot of RR – BESO - Diabetic



(d) PCM Plot of RR – BESO – Non-Diabetic

Figure 6. PCM Plots for RR FE Method.

4. Classifiers

FE acts like a filter, selecting the most informative data tidbits from the massive count of gene expressions. However, these informative features alone aren't enough to diagnose diabetes. Here's where classifiers come in as powerful tools. They analyze the selected features, learning the subtle patterns that differentiate diabetic from non-diabetic samples. These patterns can be complex and non-linear, making them difficult to identify by hand. Classifiers ultimately leverage this knowledge to make automated and accurate classifications of new, unseen diabetic cases.

4.1. Non-linear Regression

While standard linear regression assumes a straightforward, linear relationship between variables, nonlinear regression emerges as a powerful tool for diabetic detection. It tackles the challenge of complex relationships that might exist between features extracted from gene expression data and the presence of diabetes. The core principle involves modeling the relationship between extracted features and the diabetic state using a specific mathematical function that captures the non-linearity [33]. The model strives to minimize the sum of squared errors between the predicted diabetic state and the actual diabetic state for each data point. Imagine finding the best-fitting non-linear curve that represents the data.

Least squares estimation then comes into play to determine the unknown parameters within the chosen non-linear function. These parameters define the shape and characteristics of this non-linear relationship. After training the model with known diabetic and non-diabetic data points, the estimated non-linear function can predict the diabetic state for new, unseen cases based on their extracted features[34]. A threshold probability or value can be established to classify the predicted state as diabetic or non-diabetic. This approach offers a valuable technique for leveraging the complexities within gene expression data for improved diabetes detection.

$$G_m = y(a_m, \theta) + P_{mm} \quad (19)$$

Nonlinear regression centers on the concept of an expectation function, typically denoted by y . This function represents the predicted value of the dependent variable (often signifying the diabetic state) based on the independent variables (also known as features extracted from gene expression data) for each data point. Here's the key difference between nonlinear and linear models: in nonlinear regression, at least one parameter (a_m) for each data point (m) must influence one or more derivatives of the expectation function. This allows the model to capture non-linear relationships between the features and the diabetic state. Unlike linear models where the expectation function is a straight line, the nonlinear version can take on various shapes depending on the chosen mathematical formula and its parameters. This flexibility empowers the model to represent the complexities inherent in the data, potentially leading to more accurate classifications of diabetic cases.

4.2. Linear Regression

Linear regression [35], a basis of statistical analysis, excels at modeling linear relationships between variables. However, in the realm of diabetic detection, it falls short as a primary classification tool, linear regression assumes a straight-line connection between the features extracted from gene expression data (independent variables) and the diabetic state (dependent variable). This linearity might not reflect the true complexities of the underlying biology. Diabetic development can involve intricate, non-linear relationships that linear regression might miss, leading to inaccurate classifications. Next, diabetic detection is inherently a binary classification problem. While applying a threshold might seem straightforward, it can be suboptimal for capturing the nuances of the data and the disease itself.

$$Z = p + qX \quad (20)$$

4.3. Gaussian Mixture Model

Gaussian Mixture Models (GMMs) [36] shine as powerful tools for unsupervised machine learning. Unlike techniques requiring labeled data, GMMs excel at uncovering hidden structures within complex datasets like gene expression profiles. GMMs act like detectives, searching for hidden patterns in the data. They can automatically group similar gene expression profiles into clusters, potentially revealing subgroups within the diabetic or non-diabetic population. Unlike rigid clustering methods, GMMs employ a softer approach. They assign probabilities to each cluster for a given data point, allowing for potential overlap between clusters. This flexibility reflects the inherent complexities of diseases like diabetes, where gene expression patterns might not be perfectly distinct.

GMMs assume that the data arises from a blend of multiple, simpler Gaussian distributions [37]. This allows them to capture the diverse patterns present within gene expression data. By combining these Gaussian components, GMMs create a nuanced model that can effectively represent the underlying biological processes related to diabetes.

$$p(a) = \frac{1}{(2\pi)^{\frac{n}{2}} |\Sigma|^{\frac{1}{2}}} e^{-\frac{1}{2}(a-\mu)^T \Sigma^{-1} (a-\mu)} \quad (21)$$

Within the Gaussian distribution formula, the symbol μ denotes the mean vector, representing the average position of the data points in the n-dimensional space. Σ represents the covariance matrix, which captures the relationships between the different features (dimensions) within the data. The determinant of the covariance matrix, denoted by $|\Sigma|$, plays a role in calculating the overall spread of the data around the mean. Finally, the $\exp()$ function is used to calculate the probability density of a particular data point within the Gaussian distribution.

By uncovering hidden structures and capturing the complexities of gene expression data, GMMs provide valuable insights for researchers exploring novel approaches to diabetic detection.

4.4. Expectation Maximum

The EM algorithm emerges as a powerful tool for addressing missing information or hidden factors [38,39]. The EM algorithm acts like a skilled detective in diabetic detection. Two-step process such as:

- **Expectation Step:** In this initial step, the EM algorithm estimates the missing information, or hidden factors based on the currently available data and the current model parameters.
- **Maximization Step:** With the estimated missing values in place, the EM algorithm then refines the model parameters by considering the newly completed data.

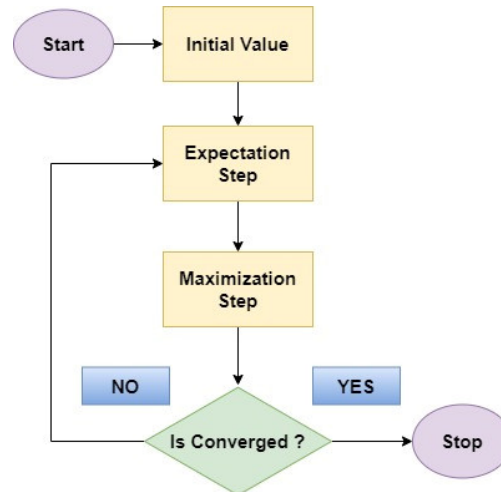


Figure 5. Flow diagram of EM algorithm.

Through this iterative process of estimating missing information and then improving the model based on those estimates, the EM algorithm gradually uncovers the underlying structure and parameters governing the distribution of gene expression data. This ultimately leads to a more accurate classification of diabetic and non-diabetic cases.

4.5. Logistic Regression

LR establishes itself as a powerful tool for binary classification tasks, proving its worth in decoding the complexities of diabetic detection. Like its role in identifying alcoholic tendencies, logistic regression [40] estimates the probability of an individual developing diabetes based on a collection of factors. The initial phase involves meticulously gathering relevant data on potential risk factors from a sample population. The collected data is then strategically divided into two distinct sets. The training set serves as the foundation for building the logistic regression model, while the testing set acts as an impartial evaluator, assessing the model's accuracy in predicting new diabetic cases. The LR model [41] meticulously analyzes the training data. It constructs a mathematical function that establishes a relationship between the collected risk factors as independent variables and the likelihood of developing diabetes as dependent variable. Once the model is fully trained and refined, it can be leveraged to predict the probability of diabetes in new individuals based solely on their specific risk factors.

$$\text{Logit}\{\Pi(x)\} = v_0 + \sum_{j=1}^q v_j x_j^* \quad (22)$$

The objective is to optimize both the fitness and log-likelihood, which can be achieved by attaining the subsequent function:

$$1(v_0, v) = \sum_{i=1}^n \{y_i \log(\pi_i) + (1 - \pi_i)\} - \frac{1}{2\tau^2} \|v\|^2 \quad (23)$$

4.6. Softmax Discriminant Classifier

SDC is an ingenious approach that leverages the power of an individual's genetic makeup to predict their potential diabetes risk. SDC [42] is meticulously analyzing a person's genes. It employs a specialized tool – a discriminant function – that acts like a comparison chart. This function compares the genetic profile against those of individuals confirmed to have or not have diabetes. The key strength of SDC lies in its ability to learn and adapt. By analyzing a set of well-defined examples (individuals with confirmed diabetic status), SDC refines its discriminant function. This refinement helps SDC sharpen the distinction between diabetic and non-diabetic gene patterns. The goal is to establish clear boundaries between these two groups, enabling accurate classification of new cases.

When presented with a new genetic profile, SDC compares it to the learned patterns. By analyzing the closest match, SDC can estimate the likelihood of an individual developing diabetes.

The process of transforming class samples and test samples in SDC incorporates nonlinear enhancement values, which are calculated utilizing the subsequent equations:

$$h(K) = \arg \max Z_w^i \tag{24}$$

$$h(K) = \arg \max_i \log \left(\sum_{j=1}^{d_i} \exp(-\lambda \|v - v_j^i\|_2) \right) \tag{25}$$

In these formulas, $h(K)$ represents the distinction of the i^{th} class, and as $\|v - v_j^i\|_2$ converges to zero, Z_w^i is maximized. This characteristic ensures that the test sample is most likely to belong to a particular class.

4.7. Support Vector Machine (Radial Basis Function)

Support Vector Machines (SVMs) are a powerful machine learning technique well-suited for classification tasks like diabetic detection. Unlike linear regression, SVMs can handle non-linear relationships between features extracted from gene expression data and the diabetic state. SVMs [43] can map the extracted features of initially existing in a lower-dimensional space into a higher-dimensional space. This transformation allows the data points to be separated more effectively using a hyperplane, even if the original relationship between features and the diabetic state was non-linear. RBF kernel is a popular choice for SVM classification. It acts like a bridge, calculating the similarity between data points in the higher-dimensional space. This similarity measure is then used by the SVM algorithm to identify the optimal hyperplane that best separates diabetic from non-diabetic data points in this transformed space. Once the optimal hyperplane is established, new, unseen data points can be mapped into the same higher-dimensional space using the RBF kernel. Their position relative to the hyperplane allows the SVM to classify them as diabetic or non-diabetic.

$$\text{RBF: } k(x_i, x_j) = \exp \left\{ \frac{-\|x_i - x_j\|^2}{(2 \cdot \sigma)^2} \right\} \tag{26}$$

By effectively handling non-linearities and leveraging the RBF kernel for similarity calculations, SVMs offer a robust approach to classifying diabetic cases based on gene expression data.

4.8. Selection of Classifiers Parameters through Training and Testing

To ensure the model’s generalizability despite limited data, data is employed a robust technique called 10-fold cross-validation. This method splits the data into 10 equal folds, trains the model on 9 folds, and tests it on the remaining fold. This process is repeated for all folds, providing a more reliable estimate of the model’s performance on unseen data compared to a single test/train split. Additionally, the model was trained on a dataset containing 2870 features per patient for 20 diabetic and 50 non-diabetic individuals. This comprehensive training process, combined with k-fold cross-validation, strengthens the reliability of the findings. The training process is measured in the MSE as

$$MSE = \frac{1}{N} \sum_{j=1}^N (O_j - T_j)^2 \tag{27}$$

O_j is the actual value observed at a specific time, and T_j is the target value the model predicts for that time.

5. Classifiers Training and Testing

Table 4 represents the confusion matrix for diabetes detection and the parameters are defined as follows,

Table 4. Confusion matrix of diabetic and non-diabetic classification.

Clinical Situation	Predicted Values	
	Dia	Non-Dia

Real Values	Class of Dia	TP	FN
	Class of non-Dia	FP	TN

TP (True Positive): Catches diabetic patients. TN (True Negative): Identifies healthy people. FP (False Positive): Mistakes a healthy person for diabetic. FN (False Negative): Misses a diabetic person.

Table 5 directly compares the Mean Squared Error (MSE) on both the training data it learns from (Train MSE) and unseen test data (Test MSE), with lower MSE indicating better performance. The highest values achieved as Support Vector Machine (SVM) with an RBF kernel, achieving exceptionally low MSE on both training and test data (Train MSE: 1.88×10^{-6} , Test MSE: 1×10^{-6}). Statistical models like Ridge Regression (Train MSE: 7.29×10^{-6} , Test MSE: 3.25×10^{-5}), Linear Regression (Train MSE: 1.16×10^{-5} , Test MSE: 1.94×10^{-5}), and Logistic Regression with L1 penalty (LoR) (Train MSE: 2.7×10^{-5} , Test MSE: 3.02×10^{-5}) also show promise with consistently low MSE values. Mixture models, Gaussian Mixture Model (GMM) (Train MSE: 1.02×10^{-5} , Test MSE: 1.48×10^{-5}) and Expectation Maximization (EM) (Train MSE: 5.29×10^{-6} , Test MSE: 1.37×10^{-5}), demonstrate competitive results. Notably, STFT and PCC have higher MSE values across both Train and Test data, suggesting they may not be as effective for this specific diabetic detection task. It's note that all models exhibit lower MSE on the training data compared to test data, indicating some degree of overfitting. This underscores the importance of evaluating models using unseen test data to ensure they generalize well to new cases.

Table 5. Mean Square Error Analysis for different FE techniques without feature selection.

Classifiers	STFT		Ridge Regression		Pearson CC	
	Train MSE	Test MSE	Train MSE	Test MSE	Train MSE	Test MSE
NLR	1.59×10^{-5}	4.84×10^{-6}	7.29×10^{-6}	3.25×10^{-5}	4.36×10^{-5}	4.1×10^{-4}
LR	1.18×10^{-5}	3.61×10^{-6}	1.16×10^{-5}	1.94×10^{-5}	9.61×10^{-6}	3.84×10^{-4}
GMM	1.05×10^{-5}	2.89×10^{-6}	1.02×10^{-5}	1.48×10^{-5}	2.02×10^{-5}	8.41×10^{-4}
EM	6.74×10^{-6}	2.89×10^{-6}	5.29×10^{-6}	1.37×10^{-5}	9.61×10^{-6}	3.72×10^{-5}
LoR	2.46×10^{-5}	9×10^{-6}	2.7×10^{-5}	3.02×10^{-5}	4×10^{-6}	2.92×10^{-5}
SDC	1.28×10^{-5}	4×10^{-6}	1.68×10^{-5}	1.22×10^{-5}	2.56×10^{-6}	1.85×10^{-5}
SVM (RBF)	1.88×10^{-6}	1×10^{-6}	2.56×10^{-6}	4.41×10^{-6}	3.6×10^{-7}	4.41×10^{-6}

In Table 6, SVM with an RBF kernel maintains its dominant position (Train MSE: 2.18×10^{-6} , Test MSE: 1.44×10^{-6}). It achieves exceptionally low MSE on both the training data it learns from and unseen test data. Statistical models show mixed results. Ridge Regression (Train MSE: 1.44×10^{-5} , Test MSE: 2.21×10^{-5}) performs consistently, while LR (Train MSE: 3.76×10^{-5} , Test MSE: 2.3×10^{-5}) delivers slightly higher training MSE but achieves a lower test MSE. Logistic Regression with L1 penalty (Train MSE: 9.97×10^{-6} , Test MSE: 1.76×10^{-5}) demonstrates the most significant improvement in test MSE compared to training MSE. Mixture models present a similar picture. GMM exhibits a significant jump in Test MSE (3.97×10^{-4}) compared to Train MSE (4.51×10^{-5}), suggesting potential overfitting. EM shows more balanced performance (Train MSE: 3.14×10^{-4} , Test MSE: 1.37×10^{-5}). Like the previous table, STFT and PCC remain less competitive in this task, evident from their higher MSE values across both Train and Test data (above 5.00×10^{-5} for most cases). An interesting observation is the significant reduction in Test MSE for SDC (Softmax Discriminant Classifier) compared to the previous table (Train MSE: 2.21×10^{-5} , Test MSE: 1.6×10^{-5} to Train MSE: 2.81×10^{-6} , Test MSE: 2.81×10^{-4}). This suggests potential improvements in model generalizability. Overall, SVM with RBF kernel remains the leader, while statistical and mixture models show varying effectiveness depending on the specific model and its ability to generalize to unseen data. The significant reduction in SDC's Test MSE warrants further investigation into its potential for diabetic detection.

Table 6. Mean Square Analysis for different Feature Extraction techniques with Bald Eagle Search Optimization (BESO) feature selection.

STFT	Ridge Regression	Pearson CC
------	------------------	------------

Classifiers	Train MSE	Test MSE	Train MSE	Test MSE	Train MSE	Test MSE
NLR	1.43×10^{-5}	5.29×10^{-5}	1.44×10^{-5}	2.21×10^{-5}	9.41×10^{-5}	7.06×10^{-5}
LR	3.76×10^{-5}	2.3×10^{-5}	7.74×10^{-5}	1.85×10^{-5}	2.5×10^{-5}	2.02×10^{-5}
GMM	4.51×10^{-5}	1.3×10^{-5}	6.56×10^{-5}	3.97×10^{-4}	6.08×10^{-5}	3.02×10^{-5}
EM	3.4×10^{-5}	1.37×10^{-5}	5.18×10^{-5}	3.14×10^{-4}	1.6×10^{-7}	1.3×10^{-5}
LoR	9.97×10^{-6}	4×10^{-6}	9×10^{-6}	1.76×10^{-5}	4.9×10^{-7}	1.68×10^{-5}
SDC	2.21×10^{-5}	1.6×10^{-5}	2.81×10^{-6}	2.81×10^{-4}	8.1×10^{-7}	8.65×10^{-5}
SVM (RBF)	2.18×10^{-6}	1.44×10^{-6}	5.29×10^{-6}	4.9×10^{-5}	4.9×10^{-7}	8.1×10^{-7}

In Table 7, SVM with an RBF kernel maintains its exceptional performance (Train MSE: 4.25×10^{-7} , Test MSE: 3.6×10^{-7}). It achieves remarkably low MSE on both the training and unseen test data, solidifying its position as the leading contender. Statistical models show further improvement. LoR (Train MSE: 4.85×10^{-5} , Test MSE: 1.96×10^{-6}) demonstrates a significant decrease in Test MSE compared to the previous table. Logistic Regression with L1 penalty (LoR) (Train MSE: 1.39×10^{-5} , Test MSE: 2.25×10^{-6}) also exhibits a positive trend with lower Test MSE. However, RR (Train MSE: 6.08×10^{-5} , Test MSE: 9×10^{-6}) shows a slight increase in Test MSE. Mixture models present a more promising picture in this table. GMM achieves a significant reduction in both Train MSE (9.01×10^{-6}) and Test MSE (4.41×10^{-6}), suggesting potential improvements in addressing overfitting. EM maintains consistent performance (Train MSE: 3.51×10^{-5} , Test MSE: 7.29×10^{-6}). Like previous observations, STFT and PCC remain less competitive (Train and Test MSE values above 5.00×10^{-6}). The most significant improvement is seen with SDC. It achieves exceptionally low Test MSE (1.96×10^{-6}) compared to both previous tables (Train MSE: 1.35×10^{-5} , Test MSE: 2.89×10^{-6}). This dramatic reduction in Test MSE warrants further investigation into the potential of SDC for diabetic detection, especially considering its consistent Train MSE across all three tables.

Table 7. Mean Square Analysis for different Feature Extraction techniques with Red Deer Optimization (RDO) feature selection.

Classifiers	STFT		Ridge Regression		Pearson CC	
	Train MSE	Test MSE	Train MSE	Test MSE	Train MSE	Test MSE
NLR	2.62×10^{-5}	2.56×10^{-6}	6.08×10^{-5}	9×10^{-6}	5.04×10^{-5}	6.56×10^{-5}
LR	4.85×10^{-5}	1.96×10^{-6}	6.24×10^{-5}	6.4×10^{-5}	2.25×10^{-6}	1.09×10^{-5}
GMM	9.01×10^{-6}	4.41×10^{-6}	2.12×10^{-5}	2.25×10^{-6}	6.25×10^{-6}	1.22×10^{-5}
EM	3.51×10^{-5}	7.29×10^{-6}	5.48×10^{-5}	2.81×10^{-5}	1.69×10^{-6}	7.84×10^{-6}
LoR	1.39×10^{-5}	2.25×10^{-6}	3.02×10^{-5}	4.84×10^{-6}	3.6×10^{-7}	4×10^{-6}
SDC	1.35×10^{-5}	2.89×10^{-6}	2.6×10^{-5}	1.96×10^{-6}	1.44×10^{-7}	1.68×10^{-5}
SVM (RBF)	4.25×10^{-7}	3.6×10^{-7}	8.1×10^{-7}	9×10^{-8}	4×10^{-8}	2.5×10^{-7}

Overall, SVM with RBF kernel remains the frontrunner. Statistical and mixture models demonstrate ongoing optimization, with Logistic Regression and GMM showing the most notable improvements. The remarkable performance boost of SDC in this table highlights its potential as a viable approach for diabetic detection.

5.1. Selection of Targets

The target value for the non-diabetic class ($T_{Non-Dia}$) ranges from 0 to 1, with emphasis on the lower end of the scale. This range is determined by a specific constraint[44].

$$\frac{1}{N} \sum_{i=1}^N \mu_i \leq T_{Non-Dia}$$

Here, μ_i denotes the mean value of the input feature vectors considered for classification across N non-diabetic features. Similarly, in the case of the diabetic class (T_{Dia}) the target value is aligned with the upper range of the 0 to 1 scale. This alignment is determined by the following principle as

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

In this context, μ_j represents the average value of input feature vectors for the M diabetic cases considered in classification. It’s crucial to note that the target value (T_{Dia}) is deliberately set higher than the average values of both μ_i and μ_j . This choice of target values mandates a minimum discrepancy between them of 0.5, as stated below,

$$||T_{Dia} - T_{Non-Dia}|| \geq 0.5$$

The target values for non-diabetic ($T_{Non-Dia}$) and diabetic (T_{Dia}) cases are set at 0.1 and 0.85, respectively. After setting the targets, MSE is employed to assess classifier performance. Table 8 illustrates the optimal parameter selection for the classifiers following the training and testing phases.

Table 8. Classifiers optimal parameters.

Classifiers	Description
NLR	The uniform weight is set to 0.4, while the bias is adjusted iteratively to minimize the sum of least square errors, with the criterion being the Mean Squared Error (MSE).
Linear Regression	The weight is uniformly set at 0.451, while the bias is adjusted to 0.003 iteratively to meet the Mean Squared Error (MSE) criterion.
GMM	The input sample’s mean covariance and tuning parameter are refined through EM steps, with MSE as the criterion.
EM	The likelihood probability is 0.13, the cluster probability is 0.45, and the convergence rate is 0.631, with the condition being MSE.
Logistic regression	The criterion is MSE, with the condition being that the threshold $H\theta(x)$ should be less than 0.48.
SDC	The parameter Γ is set to 0.5, alongside mean target values of 0.1 and 0.85 for each class.
SVM (RBF)	The settings include C as 1, the coefficient of the kernel function (gamma) as 100, class weights at 0.86, and the convergence criterion as MSE.

6. Outcomes and Findings

The evaluation approach ensures robust assessment of the models’ effectiveness in classifying diabetic patients. Researchers employ a widely used technique called “tenfold cross-validation.” Here, the data is meticulously divided into 10 equal sets. Nine of these sets (representing 90% of the data) are used to train the models. The remaining set (10%) serves a crucial purpose - testing the model’s performance on unseen data. This approach helps mitigate overfitting and provides a more reliable picture of how the models would perform in real-world scenarios. To go beyond simple accuracy, we leverage a valuable tool called a confusion matrix. This matrix allows us to calculate a comprehensive set of performance metrics. These metrics include Accuracy (overall percentage of correct predictions), F1 score (balances precision and recall), MCC (considers true positives/negatives for a more robust evaluation), Jaccard Metric (measures the shared proportion of positive predictions between the model and the ground truth), Error rate (percentage of incorrect predictions), and Kappa statistic [45] (measures agreement beyond chance). By analyzing these metrics derived from the confusion matrix, we gain a deeper understanding of the models’ strengths and weaknesses in distinguishing between diabetic and non-diabetic cases. Table 9 which are employed to calculate the performance metrics mentioned. This table provides a transparent view of the evaluation process and allows for further analysis of the models’ configuration.

Table 9. Performance metrics.

Metrics	Formula
---------	---------

Accuracy	$Accu = \frac{(TN + TP)}{(TN + FN + TP + FP)}$
F1 Score	$F1 = \frac{2 \times TP}{(2 \times TP + FP + FN)}$
Matthews Correlation Coefficient (MCC)	$MCC = \frac{(TP \times TN - FP \times FN)}{\sqrt{(TP + FP) \times (TP + FN) \times (TN + FP) \times (TN + FN)}}$
Jaccard Metric	$Jacc = \frac{TP}{TP + FP + FN}$
Error Rate	$ER = 1 - Accu$
Kappa	$Kappa = \frac{(P_o - P_e)}{(1 - P_e)}$ $P_o = \frac{(TP + TN)}{(TP + TN + FP + FN)}$ $P_e = \frac{(TP + FP) \times (TP + FN) + (FP + TN) \times (FN + TN)}{(TP + TN + FP + FN)^2}$

The result of the above metrics for evaluation of the performance in each classifier is given in the Table 9,

From Table 10, SVM with an RBF kernel solidified its dominance. It achieved the highest overall accuracy (92.8571%), F1 score (85.71%), Matthews Correlation Coefficient (MCC) (0.7979), and Jaccard similarity (75%), demonstrating its exceptional ability to distinguish between diabetic and non-diabetic cases With PCC as FE. Conversely, RR exhibited the lowest performance across most metrics, suggesting it may not be suitable for this task. While the choice of feature extraction technique appears less impactful than the model itself (except for PCC), some interesting observations emerged. The combination of STFT and RR yielded the lowest overall performance (average accuracy of 80%), indicating RR might be less effective here. Excluding these outliers (SVM and Pearson CC), the average accuracy for other models ranged from 81% to 84%, with F1 scores between 72% and 74% and MCC values between 0.60 and 0.65. Overall, SVM with an RBF kernel emerged as the strongest contender for diabetic detection due to its consistent performance across various feature extraction techniques. While other models like NLR and SDC showed promise, further investigation might be necessary to evaluate their generalizability and confirm their effectiveness.

Table 10. Analysis of different parameters with different classifiers through various FE techniques without Feature Selection techniques.

Feature Extraction	Classifiers	Parameters					
		Accu (%)	F1S (%)	MCC	Jaccard Metric (%)	Error rate (%)	Kappa
STFT	NLR	85.7142	77.2727	0.6757	62.9629	14.2857	0.6698
	LR	87.1428	79.0697	0.7021	65.3846	12.8571	0.6985
	GMM	87.1428	79.0697	0.7021	65.3846	12.8571	0.6985
	EM	87.1428	79.0697	0.7021	65.3846	12.8571	0.6985
	LoR	82.8571	72.7272	0.6091	57.1428	17.1428	0.6037
	SDC	88.5714	81.8181	0.7423	69.2307	11.4285	0.7358
	SVM (RBF)	91.4285	85.7142	0.7979	75	8.57142	0.7961
Ridge Regression	NLR	80	66.6667	0.5255	50	20	0.5242
	LR	80	68.1818	0.5425	51.7241	20	0.5377
	GMM	81.4285	71.1111	0.5845	55.1724	18.5714	0.5767
	EM	84.2857	74.4186	0.6348	59.2592	15.7142	0.6315
	LoR	71.4286	58.3333	0.3873	41.1764	28.5714	0.375
	SDC	78.5714	68.0851	0.5383	51.6129	21.4285	0.5248

	SVM (RBF)	88.5714	80.9524	0.7298	68	11.4285	0.7281
Pearson CC	NLR	65.7143	52	0.2829	35.1351	34.2857	0.2695
	LR	78.5714	65.1162	0.5001	48.2758	21.4285	0.4976
	GMM	77.1429	68	0.5385	51.5151	22.8571	0.5130
	EM	78.5714	65.1162	0.5001	48.2758	21.4285	0.4976
	LoR	82.8571	70	0.58	53.8461	17.1428	0.58
	SDC	85.7142	75	0.65	60	14.2857	0.65
	SVM (RBF)	92.8571	87.1795	0.8228	77.2727	7.14285	0.8223

From Figure 6, SVM with an RBF kernel consistently achieved the highest values across all performance metrics, demonstrating its exceptional ability to distinguish diabetic from non-diabetic cases. For instance, we observed exceptional accuracy (around 85%) and F1 score (around 80%) for SVM with RBF compared to other models. While the choice of FE technique seems to have a lesser impact on overall performance compared to the model itself (except for RR), one noteworthy observation emerged. RR, when combined with any feature extraction technique, yielded the lowest performance. Excluding RR and the outliers (SVM and Pearson CC), other models exhibited average accuracy ranging from 80% to 85%, with F1 scores between 70% and 75%. Overall, SVM with RBF kernel stands out as the most promising approach for diabetic detection based on its consistently high performance across various feature extraction techniques. While other models like NLR, SDC, and some with other feature extraction techniques show promise, further investigation might be necessary to evaluate their generalizability and confirm their effectiveness.

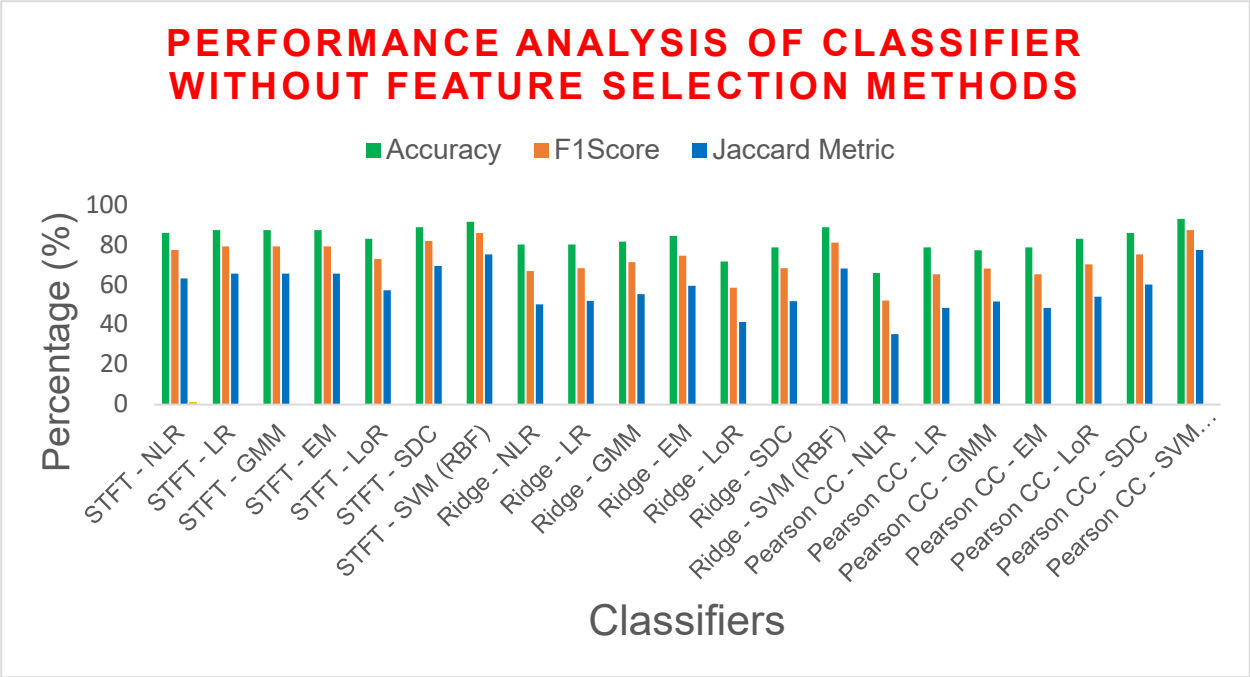


Figure 6. Performance analysis of different classifier without FS Method.

From Table 11, SVM with RBF kernel solidified its dominance. It achieved the highest overall accuracy (92.86%), F1 score (87.80%), Matthews Correlation Coefficient (MCC) (0.8280), and Jaccard similarity (78.26%), demonstrating its exceptional ability to distinguish between diabetic and non-diabetic cases with PCC as FE for BESO FS. Conversely, PCC with NLR exhibited the lowest performance across most metrics, with accuracy as low as 57.14%, F1 score of 44.44%, and MCC of 0.1446. Examining the impact of FE techniques revealed some interesting insights. The pairing of STFT with RR yielded a moderate average accuracy of 78%, indicating RR might be less effective here. However, excluding RR techniques, with STFT in NLR, LoR, GMM, EM achieved similar average accuracy (ranging from 80% to 84%) when combined with SVM (RBF) shows high as 91.42%. This highlights the consistent performance of SVM (RBF) across various FE methods. In conclusion, SVM

with an RBF kernel emerged as the strongest contender for diabetic detection based on its consistently high performance across different FE techniques.

Table 11. Analysis of different parameters with different classifiers through various FE techniques with Bald Eagle Search Optimization Feature selection techniques.

Feature Extraction	Classifiers	Parameters					
		Accu (%)	F1S (%)	MCC	Jaccard Metric (%)	Error rate (%)	Kappa
STFT	NLR	84.2857	74.4186	0.6347	59.2592	15.7142	0.6315
	LR	74.2857	65.3846	0.4987	48.5714	25.7142	0.4661
	GMM	80	69.5652	0.5609	53.3333	20	0.5504
	EM	80	69.5652	0.5609	53.3333	20	0.5504
	LoR	87.1428	79.0697	0.7021	65.3846	12.8571	0.6985
	SDC	80	70.8333	0.5809	54.8387	20	0.5625
	SVM (RBF)	91.4285	86.3636	0.8089	76	8.57142	0.8018
Ridge Regression	NLR	78.5714	66.6667	0.5185	50	21.4285	0.5116
	LR	62.8571	53.5714	0.2982	36.5853	37.1428	0.2661
	GMM	61.4285	49.0566	0.2262	32.5	38.5714	0.2092
	EM	65.7142	53.8461	0.3083	36.8421	34.2857	0.2881
	LoR	81.4285	69.7674	0.5674	53.5714	18.5714	0.5645
	SDC	71.4285	58.3333	0.3872	41.1764	28.5714	0.375
	SVM (RBF)	88.5714	82.6087	0.7573	70.3703	11.4285	0.7431
Pearson CC	NLR	57.1428	44.4444	0.1446	28.5714	42.8571	0.1322
	LR	72.8571	61.2244	0.4310	44.1176	27.1428	0.4140
	GMM	62.8571	51.8518	0.2711	35	37.1428	0.2479
	EM	91.4285	84.2105	0.7855	72.7272	8.57142	0.7835
	LoR	90	82.0512	0.7517	69.5652	10	0.7512
	SDC	81.4285	62.8571	0.5174	45.8333	18.5714	0.5081
	SVM (RBF)	92.8571	87.8048	0.8280	78.2608	7.14285	0.8275

From Figure 7, it is revealed that SVM with RBF kernel as the strongest classifier for diabetic detection, achieving exceptional performance (accuracy > 92%, F1-score > 87%, MCC > 0.82, Jaccard > 78%). Interestingly, the choice of feature extraction technique had less impact, except for Ridge Regression which yielded the lowest performance across all techniques. While other models like NLR, LR, GMM, EM, and SDC showed promise, SVM with RBF kernel emerged as the leader due to its consistent effectiveness.

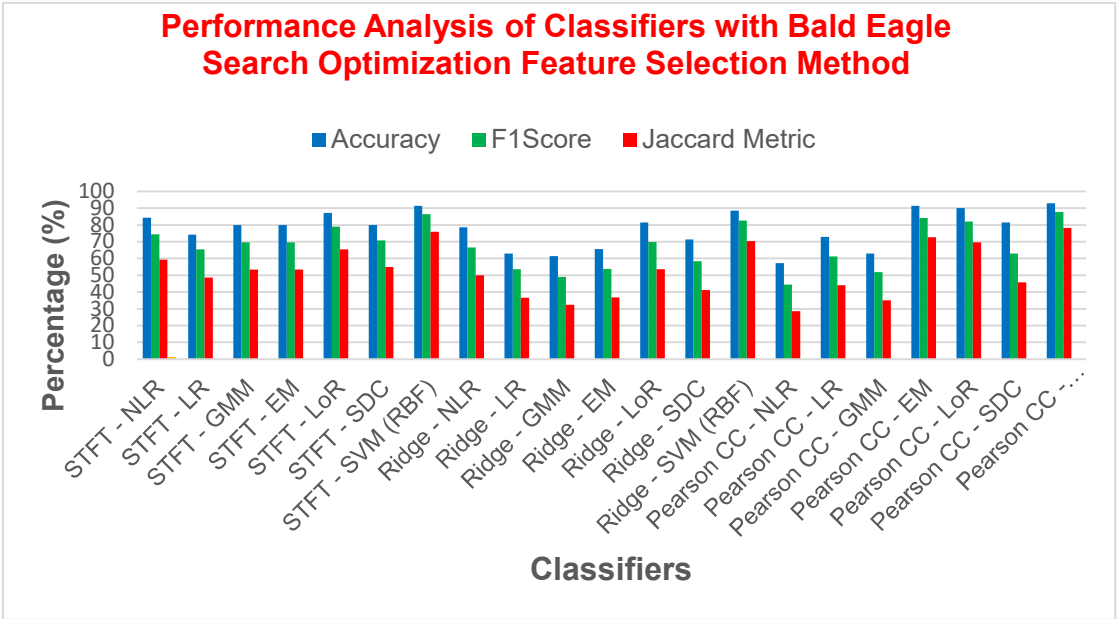


Figure 7. Performance analysis of different classifier with Bald Eagle Search Optimization FS Method.

From Table 12, it is observed that the diabetic detection models revealed SVM with RBF kernel as the champion, achieving exceptional results (accuracy reached the highest ever as 97.14%, F1-score reaches 95%, MCC exceeding 0.93, Jaccard exceeding 90%). This dominance is evident compared to the lowest performing model, Pearson CC with NLR, which obtained an accuracy as low as 62%, F1-score of 48%, and MCC of 0.22. Excluding these outliers, the average performance across models ranged from 80-85% accuracy, 70-78% F1-score, and 0.55-0.65 MCC. Interestingly, the choice of feature extraction technique seems to have a lesser impact on performance, except for Ridge Regression. It consistently yielded the lowest values (accuracy around 65%, F1-score around 55%, MCC around 0.30) regardless of the technique used. Techniques like NLR, LR, GMM, EM, LoR, and SDC performed similarly well with SVM (RBF), all achieving average accuracy above 80%. This highlights the robustness of SVM with RBF across various feature extraction approaches.

Table 12. Analysis of different parameters with different classifiers through various FE techniques with Red Deer Optimization Feature selection techniques.

Feature Extraction	Classifiers	Parameters					
		Accu (%)	F1S (%)	MCC	Jaccard Metric (%)	Error rate (%)	Kappa
STFT	NLR	90	83.7209	0.7694	72	10	0.76555
	LR	85.7142	75	0.65	60	14.2857	0.65
	GMM	88.5714	82.6087	0.7573	70.3703	11.4285	0.7431
	EM	84.2857	75.5555	0.6505	60.7142	15.7142	0.6418
	LoR	90	83.7209	0.7694	72	10	0.7655
	SDC	90	84.4444	0.7825	73.0769	10	0.7721
	SVM (RBF)	95.7142	92.6829	0.8971	86.3636	4.2857	0.8965
Ridge Regression	NLR	68.5714	60.7142	0.4248	43.5897	31.4285	0.3790
	LR	60	46.1538	0.1813	30	40	0.1694
	GMM	78.5714	70.5882	0.5820	54.5454	21.4285	0.5493
	EM	64.2857	52.8301	0.2895	35.8974	35.7142	0.2677
	LoR	74.2857	65.3846	0.4987	48.5714	25.7142	0.4661

	SDC	77.1428	69.2307	0.5622	52.9411	22.8571	0.5254
	SVM (RBF)	92.8571	88.3720	0.8367	79.1667	7.14285	0.8325
Pearson CC	NLR	62.8571	48	0.2190	31.5789	37.1428	0.2086
	LR	87.1428	78.0487	0.6901	64	12.8571	0.6896
	GMM	84.2857	74.4186	0.6347	59.2592	15.7142	0.6315
	EM	88.5714	80.9523	0.7298	68	11.4285	0.7281
	LoR	92.8571	87.1794	0.8228	77.2727	7.14285	0.8223
	SDC	80	69.5652	0.5609	53.3333	20	0.5504
	SVM (RBF)	97.1428	95	0.93	90.4761	2.85714	0.93

From Figure 8, it is observed that the analysis of diabetic detection models, SVM with RBF kernel emerged as the clear leader, achieving over 97% accuracy, with F1-score exceeding 95% and MCC surpassing 0.93. Interestingly, the choice of feature extraction method had minimal impact on performance, as models like NLR, LR, GMM, EM, LoR, and SDC performed similarly well with SVM (RBF), all achieving an average accuracy of over 80%. However, Ridge Regression consistently underperformed across all metrics, suggesting it may not be the best choice for this task. Overall, SVM with RBF kernel demonstrated exceptional and consistent performance, making it the top contender for diabetic detection.

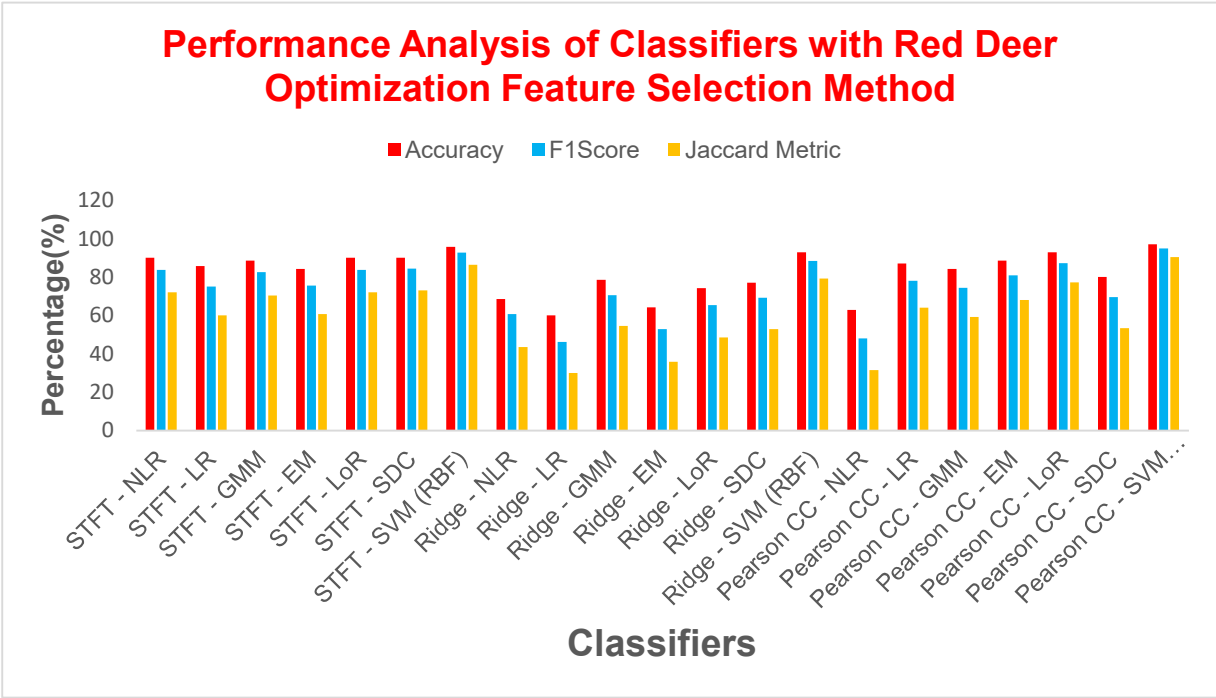


Figure 8. Performance analysis of various classifier with Red Deer Optimization FS Method.

Figure 9 illustrates the performance of the Jaccard metric and error rate (%) parameters through histograms. It is observed that the maximum error rate stabilizes at 50%, while the maximum Jaccard metric reaches 90%. The histogram representing the error rate is skewed towards the right side of the graph, indicating that regardless of the feature extraction method or feature selection method employed, the classifier’s error rate remains below 50%. On the other hand, the histogram of the Jaccard metric displays sparsity towards the edges and covers a greater number of points in the central area across the classifier.

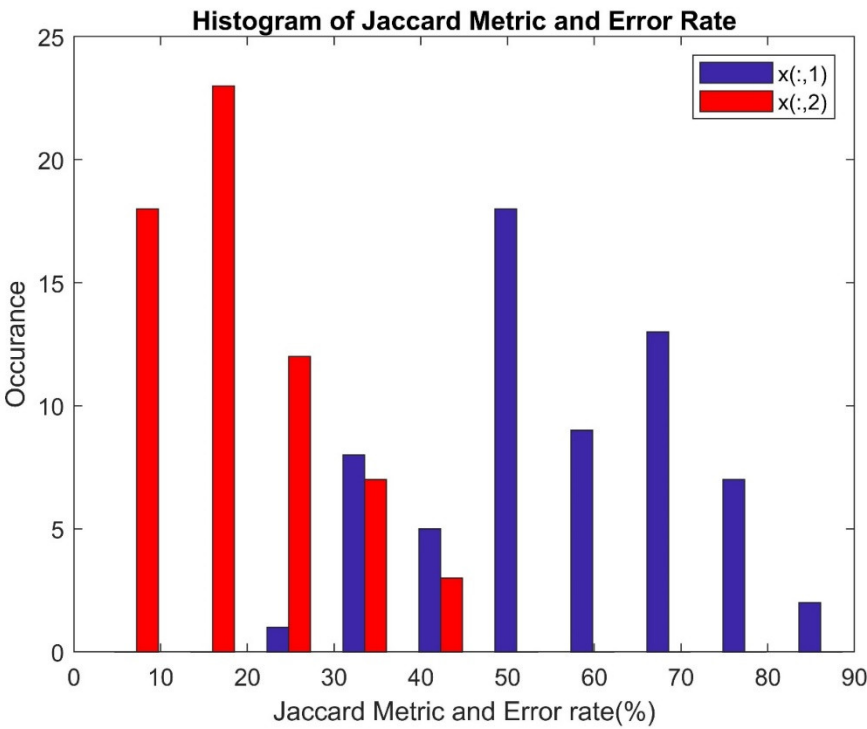


Figure 9. Performance of Jaccard Metric and Error rate.

6.1. Computational Complexity

This study evaluates classifiers based on their Computational Complexity (CC), specifically focusing on how it scales with the size of the input data (denoted as $O(n)$). Ideally, a classifier should have a low CC, represented as $O(1)$. This indicates that the algorithm’s complexity remains constant regardless of how much data it needs to process. This is a desirable characteristic because it ensures efficient performance even with large datasets. Interestingly, the analysis highlights that the CC in these models is independent of the input size, further emphasizing their efficiency. However, the text also mentions that some models exhibit a logarithmic complexity ($O(\log n)$) as the data size increases. Additionally, the study explores hybrid models that incorporate DR techniques and feature selection methods within their classification process. These techniques can potentially influence the CC of the overall model.

From Table 13, The table reveals a range of complexities across the classifiers. NLR, LR, LoR, and SDC show a complexity of $O(n^2 \log n)$ during training and then increase to $O(2n^2 \log 2n)$ or $O(2n^2 \log n)$ for prediction. GMM and EM are slightly more complex, reaching $O(n^2 \log 2n)$ in training and $O(2n^3 \log 2n)$ for prediction. However, SVM (RBF) stands out as the most computationally demanding, reaching $O(2n^4 \log 2n)$ during training and a slightly lower $O(2n^2 \log 4n)$ for prediction. In conclusion, the chosen classifier and FE technique significantly impact the computational cost of your analysis. While all classifiers show increased complexity due to the FE techniques, SVM (RBF) is the most demanding. For large datasets and a balance between efficiency and potentially good performance, consider classifiers like NLR, LR, or LoR with these FE techniques.

Table 13. Computational Complexity of different classifiers without feature selection.

Classifiers	DR Method		
	STFT	Ridge Regression	Pearson CC
NLR	$O(n^2 \log n)$	$O(2n^2 \log 2n)$	$O(2n^2 \log 2n)$
LR	$O(n^2 \log n)$	$O(2n^2 \log 2n)$	$O(2n^2 \log 2n)$

GMM	$O(n^2 \log 2n)$	$O(2n^3 \log 2n)$	$O(2n^3 \log 2n)$
EM	$O(n^3 \log n)$	$O(2n^3 \log 2n)$	$O(2n^3 \log 2n)$
LoR	$O(2n^2 \log n)$	$O(2n^2 \log 2n)$	$O(2n^2 \log 2n)$
SDC	$O(n^3 \log n)$	$O(2n^2 \log 2n)$	$O(2n^2 \log 2n)$
SVM (RBF)	$O(2n^4 \log 2n)$	$O(2n^2 \log 4n)$	$O(2n^2 \log 4n)$

Table 14 reveals a range of complexities. NLR, LR, and LoR show a complexity of $O(n^4 \log n)$ during training and then increase to $O(2n^4 \log 2n)$ for prediction. GMM and EM are slightly more complex, reaching $O(n^4 \log 2n)$ in training and $O(2n^5 \log 2n)$ for prediction. However, SVM (RBF) stands out as the most demanding, reaching a staggering $O(2n^6 \log 2n)$ during training and a slightly lower $O(2n^4 \log 4n)$ for prediction. In conclusion, both the chosen classifier and FE technique significantly impact the computational cost of your analysis. While all classifiers show increased complexity due to the FE techniques, SVM (RBF) is the most challenging.

Table 14. Computational Complexity of different classifiers with BESO features selection.

Classifiers	DR Method		
	STFT	Ridge Regression	Pearson CC
NLR	$O(n^4 \log n)$	$O(2n^4 \log 2n)$	$O(2n^4 \log 2n)$
LR	$O(n^4 \log n)$	$O(2n^4 \log 2n)$	$O(2n^4 \log 2n)$
GMM	$O(n^4 \log 2n)$	$O(2n^5 \log 2n)$	$O(2n^5 \log 2n)$
EM	$O(n^5 \log n)$	$O(2n^5 \log 2n)$	$O(2n^5 \log 2n)$
LoR	$O(2n^4 \log n)$	$O(2n^4 \log 2n)$	$O(2n^4 \log 2n)$
SDC	$O(n^5 \log n)$	$O(2n^4 \log 2n)$	$O(2n^4 \log 2n)$
SVM (RBF)	$O(2n^6 \log 2n)$	$O(2n^4 \log 4n)$	$O(2n^4 \log 4n)$

Table 15 reveals the impact of FE techniques, such as STFT, Ridge Regression, and Pearson CC, on overall complexity with RDO FS techniques. These techniques introduce additional computations, leading to a notable increase in CC across all classifiers. The table provided illustrates a spectrum of complexities for classifiers employed with these FE techniques. NLR, LR, and LoR exhibit a complexity of $O(n^5 \log n)$, while GMM and EM are slightly more complex, reaching $O(n^5 \log 2n)$. SDC demonstrates a complexity of $O(n^6 \log n)$. However, SVM (RBF) stands out as the most demanding, with a staggering complexity of $O(2n^7 \log 2n)$. Regarding feature selection, it's crucial to note that this analysis solely focuses on CC based on the chosen FE techniques. Techniques like RDO for feature selection have the potential to reduce the number of features utilized in the classification process, significantly lowering the overall complexity of the model compared to using all features.

Table 15. Computational Complexity of different classifiers with RDO features selection.

Classifiers	DR Method		
	STFT	Ridge Regression	Pearson CC
NLR	$O(n^5 \log n)$	$O(2n^5 \log 2n)$	$O(2n^5 \log 2n)$
LR	$O(n^5 \log n)$	$O(2n^5 \log 2n)$	$O(2n^5 \log 2n)$
GMM	$O(n^5 \log 2n)$	$O(2n^6 \log 2n)$	$O(2n^6 \log 2n)$
EM	$O(n^6 \log n)$	$O(2n^6 \log 2n)$	$O(2n^6 \log 2n)$
LoR	$O(2n^5 \log n)$	$O(2n^5 \log 2n)$	$O(2n^5 \log 2n)$
SDC	$O(n^6 \log n)$	$O(2n^5 \log 2n)$	$O(2n^5 \log 2n)$
SVM (RBF)	$O(2n^7 \log 2n)$	$O(2n^5 \log 4n)$	$O(2n^5 \log 4n)$

Table 16. Comparison with previous work.

S.N	Author	Description	Data	Machine	Accuracy
o	(with Year)	of the Population	Sampling	Learning Parameter	(%)

1.	This article	Nordic Islet Transplantation program	Tenfold cross-validation	STFT, RR, PCC, NLR, LR, LoR, GMM, EM, SDC, SVM(RBF)	97.14
2.	Maniruzzaman et al. (2017) [46]	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
3.	Herttroijs et al. (2018) [47]	Total: 105814 Age(mean): greater than 18	Training set of 90% and test set of 10% fivefold cross-validation	Latent Growth Mixture Modeling (LGMM)	ACC: 92.3
4.	Deo et al. (2019) [48]	Total: 140 diabetes: 14 imbalanced age: 12–90	Training set of 70% and 30% test set with fivefold cross-validation, holdout validation	BT, SVM (L)	ACC: 91
5.	Akula et al. (2019) [49]	PIDD Practice Fusion Dataset total: 10,000 age: 18–80	Training set: 800; test set: 10,000	KNN, SVM, DT, RF, GB, NN, NB	ACC: 86
6.	Xie et al. (2019) [50]	Total: 138,146 diabetes: 20,467 age: 30–80	Training set is around 67%, test set is around 33%	SVM, DT, LoR, RF, NN, NB	ACC: 81, 74, 81, 79, 82, 78
7.	Bernardini et al. (2020) [51]	Total: 252 diabetes: 252 age: 54–72	Tenfold cross-validation	Multiple instance learning boosting	ACC: 83
8.	Zhang et al. (2021) [52]	Total: 37,730 diabetes: 9.4% age: 50–70 imbalanced	Training set is around 80% test set is around 20% Tenfold cross-validation	Bagging boosting, GBT, RF, GBM	ACC: 82

6.2. Limitations

Investigated the potential of using microarray gene data to identify type 2 diabetes early and potentially predict associated diseases. The analysis revealed promising classification techniques that could be valuable for screening and identifying diabetes markers, along with potentially linked diseases like strokes and kidney problems. While the findings may be specific to this patient group and require further validation, they pave the way for future research. The methods used, such as microarrays, might not be readily available in all settings due to cost and complexity. However, this study lays the groundwork for developing more accessible and efficient approaches for early diabetes detection and disease management. Overall, the research highlights the potential for early detection of type 2 diabetes and associated diseases, emphasizing the importance of further research to validate these findings and develop more accessible screening methods for improved patient outcomes.

6.3. Conclusions and Future Work

This Analysis explored various FE techniques (STFT, Ridge Regression, and PCC) and feature selection methods (BESO and RDO) for their impact on classifier performance in detecting Type II DM using microarray gene data. While RR with RDO FS yielded lower accuracy, STFT and PCC showcased improved performance metrics, particularly for the SVM (RBF) classifier. Notably, the combination of SVM (RBF) and RDO achieved the highest accuracy (92.85%) even without feature selection. However, employing RDO alongside SVM (RBF) still resulted in the highest accuracy across all FE techniques (95%, 92%, and 97.14% for STFT, RR, and PCC respectively). BESO also

yielded promising results, achieving high accuracy values (around 90%) with all three FE techniques. Interestingly, computational complexity remained similar across classifiers with different dimensionality reduction methods, highlighting the crucial role of feature selection in boosting accuracy. In conclusion, this study presents a novel approach for Type II DM detection using microarrays. Future research will explore the application of Convolutional Neural Networks (CNNs), Deep Learning Networks (DNNs), and Long Short-Term Memory (LSTM) networks, along with hyperparameter tuning for further performance optimization.

References

1. Saeedi, P., Petersohn, I., Salpea, P., Malanda, B., Karuranga, S., Unwin, N., ... & IDF Diabetes Atlas Committee. (2019). Global and regional diabetes prevalence estimates for 2019 and projections for 2030 and 2045: Results from the International Diabetes Federation Diabetes Atlas. *Diabetes research and clinical practice*, 157, 107843.
2. Mohan, V., Sudha, V., Shobana, S., Gayathri, R., & Krishnaswamy, K. (2023). Are unhealthy diets contributing to the rapid rise of type 2 diabetes in India?. *The Journal of Nutrition*, 153(4), 940-948.
3. Oberoi, S., & Kansra, P. (2020). Economic menace of diabetes in India: a systematic review. *International journal of diabetes in developing countries*, 40, 464-475.
4. Chellappan, D., & Rajaguru, H. (2023). Detection of Diabetes through Microarray Genes with Enhancement of Classifiers Performance. *Diagnostics*, 13(16), 2654.
5. Gowthami, S., Reddy, R. V. S., & Ahmed, M. R. (2024). Exploring the effectiveness of machine learning algorithms for early detection of Type-2 Diabetes Mellitus. *Measurement: Sensors*, 31, 100983.c
6. Tasin, I., Nabil, T. U., Islam, S., & Khan, R. (2023). Diabetes prediction using machine learning and explainable AI techniques. *Healthcare technology letters*, 10(1-2), 1-10.
7. Frasca, M., La Torre, D., Pravettoni, G., & Cutica, I. (2024). Explainable and interpretable artificial intelligence in medicine: a systematic bibliometric review. *Discover Artificial Intelligence*, 4(1), 15.
8. Chaddad, A., Peng, J., Xu, J., & Bouridane, A. (2023). Survey of explainable AI techniques in healthcare. *Sensors*, 23(2), 634.
9. Hussain, F., Hussain, R., & Hossain, E. (2021). Explainable artificial intelligence (XAI): An engineering perspective. *arXiv preprint arXiv:2101.03613*.
10. Markus, A. F., Kors, J. A., & Rijnbeek, P. R. (2021). The role of explainability in creating trustworthy artificial intelligence for health care: a comprehensive survey of the terminology, design choices, and evaluation strategies. *Journal of biomedical informatics*, 113, 103655.
11. Prajapati, S., Das, H., & Gourisaria, M. K. (2023). Feature selection using differential evolution for microarray data classification. *Discover Internet of Things*, 3(1), 12.
12. Alsattar, H. A., Zaidan, A. A., & Zaidan, B. B. (2020). Novel meta-heuristic bald eagle search optimisation algorithm. *Artificial Intelligence Review*, 53, 2237-2264.
13. Hilal, A. M., Alrowais, F., Al-Wesabi, F. N., & Marzouk, R. (2023). Red Deer Optimization with Artificial Intelligence Enabled Image Captioning System for Visually Impaired People. *Computer Systems Science & Engineering*, 46(2).
14. Horng, J. T., Wu, L. C., Liu, B. J., Kuo, J. L., Kuo, W. H., & Zhang, J. J. (2009). An expert system to classify microarray gene expression data using gene selection by decision tree. *Expert Systems with Applications*, 36(5), 9072-9081.
15. Shaik, B. S., Naganjaneyulu, G. V. S. S. K. R., Chandrasheker, T., & Narasimhadhan, A. V. (2015). A method for QRS delineation based on STFT using adaptive threshold. *Procedia Computer Science*, 54, 646-653.
16. Bar, N., Nikparvar, B., Jayavelu, N. D., & Roessler, F. K. (2022). Constrained Fourier estimation of short-term time-series gene expression data reduces noise and improves clustering and gene regulatory network predictions. *BMC bioinformatics*, 23(1), 330.
17. Imani, M., & Ghassemian, H. (2015). Ridge regression-based feature extraction for hyperspectral data. *International Journal of Remote Sensing*, 36(6), 1728-1742.
18. Paul, S., & Drineas, P. (2016). Feature selection for ridge regression with provable guarantees. *Neural computation*, 28(4), 716-742.
19. Prabhakar, S. K., Rajaguru, H., Ryu, S., Jeong, I. C., & Won, D. O. (2022). A holistic strategy for classification of sleep stages with EEG. *Sensors*, 22(9), 3557.
20. Mehta, P., Bukov, M., Wang, C. H., Day, A. G., Richardson, C., Fisher, C. K., & Schwab, D. J. (2019). A high-bias, low-variance introduction to machine learning for physicists. *Physics reports*, 810, 1-124.
21. Li, G., Zhang, A., Zhang, Q., Wu, D., & Zhan, C. (2022). Pearson correlation coefficient-based performance enhancement of broad learning system for stock price prediction. *IEEE Transactions on Circuits and Systems II: Express Briefs*, 69(5), 2413-2417.
22. Mu, Y., Liu, X., & Wang, L. (2018). A Pearson's correlation coefficient-based decision tree and its parallel implementation. *Information Sciences*, 435, 40-58.

23. Grace Elizabeth Rani, T. G., & Jayalalitha, G. (2016). Complex patterns in financial time series through Higuchi's fractal dimension. *Fractals*, 24(04), 1650048.
24. Rehan, I., Rehan, K., Sultana, S., & Rehman, M. U. (2024). Fingernail Diagnostics: Advancing type II diabetes detection using machine learning algorithms and laser spectroscopy. *Microchemical Journal*, 110762.
25. Alsattar, H. A., Zaidan, A. A., & Zaidan, B. B. (2020). Novel meta-heuristic bald eagle search optimisation algorithm. *Artificial Intelligence Review*, 53, 2237-2264.
26. Kwakye, B. D., Li, Y., Mohamed, H. H., Baidoo, E., & Asenso, T. Q. (2024). Particle guided metaheuristic algorithm for global optimization and feature selection problems. *Expert Systems with Applications*, 248, 123362.
27. Wang, J., Ouyang, H., Zhang, C., Li, S., & Xiang, J. (2023). A novel intelligent global harmony search algorithm based on improved search stability strategy. *Scientific Reports*, 13(1), 7705.
28. Fard AF, Hajiaghahi-Keshteli M. Red Deer Algorithm (RDA); a new optimization algorithm inspired by Red Deers' mating. In: International conference on industrial engineering. IEEE; 2016, p. 331-42, 12. (2016).
29. Fathollahi-Fard, et al. Red deer algorithm (RDA): a new nature-inspired meta-heuristic. *Soft Comput* 2020;24.19:14637-65, (2020).
30. Bektaş, Y., & Karaca, H. (2022). Red deer algorithm based selective harmonic elimination for renewable energy application with unequal DC sources. *Energy Reports*, 8, 588-596.
31. Kumar, A. P., & Valsala, P. (2013). Feature Selection for high Dimensional DNA Microarray data using hybrid approaches. *Bioinformation*, 9(16), 824.
32. Zhang, G.; Allaire, D.; Cagan, J. Reducing the Search Space for Global Minimum: A Focused Regions Identification Method for Least Squares Parameter Estimation in Nonlinear Models. *J. Comput. Inf. Sci. Eng.* 2023, 23, 021006.
33. Draper, N.R.; Smith, H. In *Applied Regression Analysis*; John Wiley & Sons: Hoboken, NJ, USA, 1998; Volume 326.
34. Zhang, G.; Allaire, D.; Cagan, J. Reducing the Search Space for Global Minimum: A Focused Regions Identification Method for Least Squares Parameter Estimation in Nonlinear Models. *J. Comput. Inf. Sci. Eng.* 2023, 23, 021006.
35. Prabhakar, S.K.; Rajaguru, H.; Lee, S.-W. A comprehensive analysis of alcoholic EEG signals with detrend fluctuation analysis and post classifiers. In *Proceedings of the 2019 7th International Winter Conference on Brain-Computer Interface (BCI)*, Gangwon, Republic of Korea, 18-20 February 2019.
36. Llahi, O.; Rista, A. Prediction and Detection of Diabetes using Machine Learning. In *Proceedings of the 20th International Conference on Real-Time Applications in Computer Science and Information Technology (RTA-CSIT)*, Tirana, Albania, 21-22 May 2021; pp. 94-102.
37. Hamid, I.Y. Prediction of Type 2 Diabetes through Risk Factors using Binary Logistic Regression. *J. Al-Qadisiyah Comput. Sci. Math.* 2020, 12, 1.
38. 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.
39. Moon, T. K. (1996). The expectation-maximization algorithm. *IEEE Signal processing magazine*, 13(6), 47-60.
40. Adiwijaya, K.; Wisesty, U.N.; Lisnawati, E.; Aditsania, A.; Kusumo, D.S. Dimensionality reduction using principal component analysis for cancer detection based on microarray data classification. *J. Comput. Sci.* 2018, 14, 1521-1530.
41. Peng, C. Y. J., Lee, K. L., & Ingersoll, G. M. (2002). An introduction to logistic regression analysis and reporting. *The journal of educational research*, 96(1), 3-14.
42. Zang, F.; Zhang, J.S. Softmax Discriminant Classifier. In *Proceedings of the 3rd International Conference on Multimedia Information Networking and Security*, Shanghai, China, 4-6 November 2011; pp. 16-20.
43. Yao, X.; Panaye, A.; Doucet, J.; Chen, H.; Zhang, R.; Fan, B.; Liu, M.; Hu, Z. Comparative classification study of toxicity mechanisms using support vector machines and radial basis function neural networks. *Anal. Chim. Acta* 2005, 535, 259-273.
44. Ortiz-Martínez, M., González-González, M., Martagón, A. J., Hlavinka, V., Willson, R. C., & Rito-Palomares, M. (2022). Recent developments in biomarkers for diagnosis and screening of type 2 diabetes mellitus. *Current diabetes reports*, 22(3), 95-115.
45. Maxwell, A. E., Warner, T. A., & Guillén, L. A. (2021). Accuracy assessment in convolutional neural network-based deep learning remote sensing studies—Part 1: Literature review. *Remote Sensing*, 13(13), 2450.
46. 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.
47. 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.

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.
49. Akula R, Nguyen N, Garibay I. Supervised machine learning based ensemble model for accurate prediction of type 2 diabetes. In: 2019 Southeast Con. 2019.
50. Xie Z, Nikolayeva O, Luo J, Li D. Building risk prediction models for type 2 diabetes using machine learning techniques. *Prev Chronic Dis*. 2019.
51. 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.
52. 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.

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