bioengineering

MDPI

Article

Metaheuristic Class Balancing and Hyperparameter Optimization for Diabetes Classification: A Two-Stage PSO–GA Framework

Muhammad Ali Abid [1], Muhammad Irfan Saeed [1], Farhan Amin [2,*], Isabel de la Torre [3], Mirtha Silvana Garat de Marin [4] and Eduardo Silva Alvarado [5]

[1] Department of Artificial Intelligence, University of Agriculture, Dera Ismail Khan 29050, Pakistan; ali.abid@uad.edu.pk (M.A.A.); irfan.awan@uad.edu.pk (M.I.S.)
[2] School of Computer Science and Engineering, Yeungnam University, Gyeongsan 38541, Republic of Korea
[3] eHealth and Telemedicine Group, University of Valladolid, 47011 Valladolid, Spain; isator@uva.es
[4] Department of Projects, Universidad Internacional Iberoamericana, Arecibo, PR 00613, USA; silvana.mri@unic.co.ao
[5] Department of Projects, Universidad Europea del Atlántico, Isabel Torres 21, 39011 Santander, Spain; eduard.silv@uneatlantico.es
* Correspondence: farhanamin10@hotmail.com

Abstract

Background/Objectives: Accurate identification of individuals with diabetes or prediabetes from population health data can support screening and population-level disease management. However, substantial class imbalance in health datasets can bias machine learning classifiers toward the majority class and reduce the detection of affected individuals. This study proposes a two-stage metaheuristic machine learning framework combining Particle Swarm Optimization (PSO)-based class balancing with Genetic Algorithm (GA)-based classifier optimization for diabetes classification using health, behavioral, demographic, and socioeconomic information. Methods: The proposed framework integrates two optimization stages. First, Particle Swarm Optimization (PSO) is used to select majority-class training observations to address class imbalance. Second, a Genetic Algorithm (GA) is applied to optimize the hyperparameters of three classifiers: Multilayer Perceptron (MLP), implemented as an Artificial Neural Network (ANN); Extreme Gradient Boosting (XGBoost) version 3.4.1; and Random Forest (RF). The framework is evaluated using the CDC Diabetes Health Indicators dataset, comprising 253,680 observations and 21 predictors. Predictive performance is compared across the original imbalanced data, random undersampling, and PSO-based balancing using Accuracy, Specificity, ROC-AUC, and average precision. Results: The proposed PSO–GA framework demonstrates strong discrimination between individuals with and without diabetes/prediabetes. The best-performing model achieves a ROC-AUC of approximately 0.95, an average precision of 0.87, and an accuracy of 0.90 in the reported evaluation. These findings suggest that combining optimization-based class balancing with evolutionary hyperparameter tuning can support predictive modeling in imbalanced population health datasets. Conclusions: The proposed two-stage PSO–GA framework provides a systematic approach to diabetes classification by integrating majority-class sample selection with hyperparameter optimization of ANN, XGBoost, and RF classifiers. The findings demonstrate the potential of metaheuristic optimization for improving classification performance in imbalanced health datasets. Further validation using independent populations and prospective healthcare data is required before clinical deployment.

Check for updates

Academic Editors: Julfikar Haider, Sang Won Park and Mominul Ahsan

Received: 17 September 2026
Revised: 4 November 2026
Accepted: 10 November 2026
Published: 13 November 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

Keywords: diabetes risk prediction; machine learning; particle swarm optimization; genetic algorithm; class imbalance; hyperparameter optimization; healthcare analytics; predictive modeling; decision support; socioeconomic determinants

1. Introduction

Diabetes mellitus is a chronic metabolic disorder characterized by impaired insulin secretion, reduced insulin sensitivity, or a combination of both. Its prevalence has increased substantially worldwide, with hundreds of millions of individuals currently affected [1,2]. This growing burden places considerable pressure on healthcare systems and highlights the need for reliable approaches that can identify individuals at elevated risk before severe complications develop. Earlier risk identification can support preventive interventions, more appropriate clinical follow-up, and improved disease management. Machine learning (ML) has increasingly been explored as a tool for diabetes prediction because of its ability to identify complex relationships among clinical, behavioral, demographic, and socioeconomic variables [3]. Numerous studies have applied conventional and advanced ML algorithms to diabetes detection and risk assessment [4–8]. Diabetes is generally classified into type 1, type 2, and gestational diabetes. Type 1 diabetes is primarily associated with the loss of pancreatic β-cell function, whereas type 2 diabetes is characterized mainly by insulin resistance and progressive impairment of insulin secretion. Gestational diabetes develops during pregnancy and is associated with increased health risks for both the mother and child [9–11]. Clinically, diabetes is diagnosed using established glycemic criteria, including fasting plasma glucose thresholds, and is associated with serious long-term complications such as cardiovascular and cerebrovascular disease [12]. Consequently, timely identification of individuals at increased risk is important for reducing disease progression and preventing complications [13]. However, large-scale clinical screening can be costly and resource-intensive, particularly in populations where access to laboratory testing and specialist healthcare is limited. Predictive models based on routinely available health, lifestyle, and socioeconomic information may therefore provide a useful complementary approach for identifying individuals who could benefit from further clinical evaluation. Despite substantial progress in ML-based diabetes prediction, several methodological limitations remain. One of the most important is class imbalance, which occurs when non-diabetic individuals substantially outnumber diabetic individuals in the available dataset [14–17]. Models trained directly on such data may become biased toward the majority class. As a result, they can produce apparently high overall Accuracy while performing poorly in identifying individuals belonging to the clinically important minority class [17]. Several strategies have been proposed to address this problem. Random undersampling reduces the number of majority-class observations, whereas synthetic oversampling methods such as the Synthetic Minority Oversampling Technique (SMOTE) generate additional minority-class samples. Although these approaches can reduce imbalance, they also have limitations. Undersampling may remove informative observations, while synthetic oversampling may introduce artificial samples that do not fully represent the underlying population distribution [18,19]. In addition, resampling methods are often applied using predefined rules rather than being optimized according to an explicit predictive objective. Model configuration represents a second important challenge. The performance of classifiers such as Support Vector Machines and Gradient-Boosted Decision Trees can vary substantially according to the selected hyperparameters. Default settings may not be appropriate for a particular dataset, whereas exhaustive search methods can become computationally inefficient as the search space increases [20,21]. Consequently, effective diabetes prediction requires not only

appropriate treatment of class imbalance but also systematic optimization of model parameters. Most existing studies address these two problems independently. Class balancing is typically performed as a preprocessing step, followed by classifier training and separate hyperparameter tuning. This sequential but independently configured process may fail to identify a data distribution and model configuration that jointly maximize predictive performance. To address this limitation, the present study proposes a two-stage metaheuristic framework that integrates Particle Swarm Optimization (PSO) for training data balancing with Genetic Algorithm (GA)-based hyperparameter optimization.

The framework applies PSO to select majority-class training observations and GA to optimize the hyperparameters of the ANN/MLP, XGBoost, and Random Forest classifiers. These stages address class imbalance and model configuration, respectively.

The proposed framework is evaluated using the publicly available Diabetes Health Indicators dataset, which contains health-related, behavioral, demographic, and socioeconomic variables together with diabetes status labels. The dataset provides an appropriate test environment because of its large sample size and substantial class imbalance.

The main contributions of this study are as follows:

- A two-stage PSO–GA metaheuristic framework is proposed for diabetes risk prediction, in which PSO addresses training data imbalance, and the GA optimizes classifier hyperparameters.
- The PSO-based balancing strategy is designed to reduce majority-class bias while preserving informative observations, providing an optimization-driven alternative to conventional random undersampling.
- The framework evaluates multiple machine learning classifiers under a unified optimization pipeline to determine how data balancing and hyperparameter tuning jointly influence predictive performance.
- The proposed approach is assessed using accuracy, aunder the receiver operating characteristic curve (ROC-AUC), and average precision, with comparisons against baseline configurations to examine its effectiveness for imbalanced diabetes prediction.

The remainder of this paper is organized as follows. Section 2 reviews related studies on machine learning-based diabetes prediction and imbalance handling. Section 3 describes the proposed PSO–GA framework and its implementation. Section 4 presents the simulation, experimental design, evaluation procedure, and achived results. Finally, Section 5 summarizes the principal findings, limitations, and directions for future research. The Section 6 concludes the conclusion of our research.

2. Related Work

Machine learning techniques have been extensively investigated for diabetes prediction using clinical measurements, lifestyle indicators, demographic characteristics, and population health information. Previous studies have employed conventional classifiers, neural networks, and ensemble learning approaches to identify individuals with diabetes or an elevated risk of developing the disease. Naz and Ahuja investigated deep learning techniques for diabetes prediction using the Pima Indian Diabetes dataset, while Bhat et al. presented a machine learning framework for diabetes risk assessment. G In [22] explored ensemble learning methods for predicting type 2 diabetes using lifestyle indicators, highlighting the importance of behavioral and health-related variables. Similarly, in [23] investigated machine learning approaches for identifying potentially misdiagnosed adult-onset type 1 diabetes, demonstrating the relevance of predictive modeling to diabetes-related clinical assessment. Other studies have examined diabetes prediction in specific populations and clinical settings. In [24], the authors investigated postpartum prediabetes prediction among women with gestational diabetes, while in [25] the authors compared machine

learning algorithms for diabetes prediction and prevention. The authors in [26] evaluated machine learning methods against conventional logistic regression for gestational diabetes prediction. These investigations demonstrate the applicability of machine learning across different diabetes-related prediction tasks [27]. However, differences in dataset characteristics, population composition, and evaluation procedures limit direct comparisons between studies. Class imbalance remains an important methodological challenge in diabetes classification, particularly when observations from individuals without diabetes substantially outnumber those from individuals with diabetes or prediabetes. Under such conditions, conventional classifiers may favor the majority class, producing high overall accuracy while exhibiting reduced sensitivity toward minority-class observations. Common approaches for addressing this problem include random undersampling, random oversampling, and the Synthetic Minority Oversampling Technique (SMOTE) [28]. Although these methods can improve class representation, random undersampling may discard informative observations, while synthetic oversampling can alter the composition of the training dataset. These limitations motivate the investigation of more selective, optimization-based sampling approaches. Hyperparameter optimization represents another important consideration in machine learning-based diabetes prediction. The predictive performance of Artificial Neural Networks (ANNs), Extreme Gradient Boosting (XGBoost), and Random Forest (RF) depends on model-specific parameters, including network architecture, learning rate, ensemble size, tree depth, and regularization settings [29]. Conventional grid-search and random-search methods provide systematic optimization strategies but may become computationally expensive when the number of candidate configurations increases. Evolutionary optimization techniques offer an alternative by exploring candidate solutions through iterative selection and variation. Particle Swarm Optimization (PSO), originally introduced by Kennedy and Eberhart, is a population-based optimization technique that updates candidate solutions using individual and collective search experience. In classification applications, PSO can be formulated as an instance selection procedure in which particles represent candidate subsets of training observations. Genetic Algorithms (GAs), in contrast, employ evolutionary mechanisms such as selection, crossover, and mutation to identify suitable parameter configurations [30]. The complementary characteristics of these approaches motivate their integration into a sequential optimization framework. The CDC Diabetes Health Indicators dataset provides a relevant setting for investigating these challenges because it contains 253,680 observations and 21 health-related predictors covering behavioral, demographic, socioeconomic, and general health characteristics. Its large sample size and imbalanced class distribution support the investigation of data balancing strategies and classifier optimization [31]. Nevertheless, the effectiveness of optimization-based instance selection must be evaluated carefully because selective undersampling may alter the original majority-class distribution and influence classifier generalization. Although previous studies have demonstrated the usefulness of individual machine learning classifiers and imbalance-handling techniques, the present study focuses on the sequential integration of optimization-driven majority-class instance selection and evolutionary hyperparameter tuning [32,33]. Accordingly, a two-stage PSO–GA framework is investigated in which PSO is applied to majority-class instance selection, and a GA is employed to optimize the hyperparameters of ANN/MLP, XGBoost, and RF classifiers [34]. The framework is evaluated using accuracy, ROC-AUC, and average precision to examine its predictive performance on the CDC Diabetes Health Indicators dataset. In [35], the authors proposed a framework for early prediction of Schistosomiasis. The proposed a framework by incorporating the most significant features to predict Schistosomiasis using machine learning techniques.Table 1 summarizes selected previous studies and highlights their contributions and limitations in relation to the proposed framework.

Table 1. Summary of recent diabetes prediction studies and their limitations.

Ref.	Study	Dataset	Method	Key Contribution	Limitation
[2]	Parkhi et al. (2023)	Gestational Diabetes Dataset	Machine Learning Models	Predicted postpartum prediabetes in women with gestational diabetes using ML techniques.	Developed on a specific gestational diabetes cohort, limiting generalizability to the broader diabetic population.
[3]	Islam et al. (2023)	Diabetes Dataset	Comparative ML Models	Compared multiple machine learning algorithms for diabetes prediction and prevention.	Focused mainly on classifier comparison without incorporating feature optimization, hyperparameter tuning, or class imbalance handling.
[4]	Belsti et al. (2023)	Monash GDM Dataset	ML and Logistic Regression	Evaluated machine learning methods against traditional logistic regression for gestational diabetes prediction.	Limited to a specific population and lacked validation on large and diverse datasets.
[20]	Ganie and Malik (2022)	Lifestyle-Based Dataset	Ensemble Learning	Improved type II diabetes prediction using lifestyle indicators and ensemble learning techniques.	Did not address feature redundancy or employ advanced optimization strategies to enhance model performance.
[21]	Cheheltani et al. (2022)	Adult-Onset Diabetes Dataset	Machine Learning Models	Identified misdiagnosed adult-onset type 1 diabetes patients using clinical features.	Focused on diabetes subtype classification rather than general diabetes risk prediction and lacked large-scale population validation.
Proposed Study	CDC Diabetes Health Indicators Dataset	PSO-GA + RF/XGB/ANN	Integrates correlation-based predictor filtering, PSO-based majority-class instance selection, and GA-based hyperparameter optimization.	Addresses feature redundancy, class imbalance, and parameter optimization while improving scalability and generalization on a large population dataset.	

3. Materials and Methods

In this section, we present the dataset, and later we explain our proposed framework.

3.1. Dataset Description and Class Distribution

The CDC Diabetes Health Indicators dataset comprises 253,680 observations and 21 predictor variables describing health status, lifestyle, demographic characteristics, and socioeconomic factors. The dataset is derived from the Behavioral Risk Factor Surveillance System (BRFSS). The analysis concerns classification of survey-defined diabetes status. The exact handling of prediabetic observations and the corresponding class counts must be verified against the dataset and target-label encoding used in the experiments before final submission.

3.2. PSO-Based Majority-Class Instance Selection for Diabetes Data Balancing

Algorithm 1 presents the PSO procedure used for majority-class instance selection. The objective is to reduce training class imbalance by retaining a fixed-size subset of majority-class observations.

Algorithm 1. PSO-Based Majority-Class Instance Selection for Diabetes Data Balancing

Input: D_maj, D_min, target size k, swarm size P, iterations T_max, inertia w, coefficients c_1 and c_2.
Output: Balanced training dataset D_balanced.

1. Compute the minority-class centroid:
   $\mu_min = (1/|D_min|)\ \Sigma_\{z \in D_min\}\ z$
2. Initialize positions $x_p{}^0$ and velocities $v_p{}^0$ for particles p = 1, ..., P.
3. Convert each particle position into a binary selection vector s_p with exactly k selected majority-class records.
4. Evaluate each particle using the objective function:
   $F_PSO(s_p) = (1/k)\ \Sigma_{i=1}{}^M\ s_{p,i}\ \|a_i - \mu_min\|_2$
   subject to $\Sigma_{i=1}{}^M\ s_{p,i} = k$ and $s_{p,i} \in \{0, 1\}$; M = |D_maj|.
5. Initialize each particle's personal best ($pbest_p$) and the global best (gbest).
6. FOR t = 0, 1, ..., T_max − 1 DO
   a. Update velocity:
   $v_p{}^{t+1} = wv_p{}^t + c_1r_1(pbest_p - x_p{}^t) + c_2r_2(gbest - x_p{}^t)$
   b. Update position:
   $x_p{}^{t+1} = x_p{}^t + v_p{}^{t+1}$
   c. Convert updated positions into feasible binary selection vectors containing exactly k selected records.
   d. Evaluate F_PSO for each updated selection vector.
   e. Update $pbest_p$ and gbest whenever a lower fitness value is found.
   f. IF the convergence criterion is satisfied, BREAK.
7. END FOR
8. Let s* denote the binary selection vector associated with gbest.
9. Construct the balanced training dataset:
   $D_balanced = D_min \cup \{a_i \in D_maj : s^*_i = 1\}$
10. RETURN D_balanced.

In the proposed approach, each particle represents a candidate subset of majority-class instances. The position of particle (i) is defined as follows:

$$\left[\ X_i = \left[x_{\{i1\}}, x_{\{i2\}}, , x_{\{in\}}\right],\ \right] \tag{1}$$

where $(x_\{ij\}\{0,1\})$ indicates whether the (j)-th majority-class sample is selected (1) or excluded (0). Thus, each particle encodes a potential balanced subset of the majority class.

The fitness of each particle is evaluated based on the similarity between the selected majority-class samples and the minority-class distribution. Specifically, the fitness function minimizes the average Euclidean distance between the selected majority samples and the centroid of the minority class:

$$\left[\ F(X_i) = {}_{\{k=1\}}^{\{N_s\}} d(k, \{min\})\ \right], \tag{2}$$

where (Ns) denotes the number of selected majority-class samples, (k) represents the feature vector of the (k)-th selected sample, and $(\{min\})$ is the centroid of the minority class. Lower fitness values indicate a more representative subset and a better-balanced dataset.

The Euclidean distance between two d-dimensional feature vectors $\mathbf{x} = (x_1, x_2, \ldots, x_d))$ and $\mathbf{y} = (y_1, y_2, \ldots, y_d))$ is computed as

$$d_E(\mathbf{x}, \mathbf{y}) = \sqrt{\sum_{i=1}^{d} (x_i - y_i)^2} \quad (3)$$

where $d_E(\mathbf{x}, \mathbf{y})$ represents the Euclidean distance between the two feature vectors, x_i and y_i denote the values of the i th feature in vectors x and y. In addition, d indicates the total number of features. The Euclidean distance quantifies the straight-line distance between two points in a d dimensional feature space, with smaller values indicating greater similarity between the feature vectors in terms of their Euclidean proximity.

The velocity and position of each particle are updated according to the standard PSO equations:

$$\left[v_{i\{t+1\}} = wv_{it} + c_{1r_{1(pbest_i - x_{it})}} + c_{2r_{2(gbest - x_{it})}} \right], \quad (4)$$

$$\left[x_{i\{t+1\}} = x_{it} + v_i^{\{t+1\}} \right], \quad (5)$$

where (w) is the inertia weight, (c_1) and (c_2) are the cognitive and social acceleration coefficients, respectively, and $(\mathrm{r_1})$ and $(\mathrm{r_2})$ are random values uniformly distributed in the interval $[0, 1])$.

The terms (pbest_i) and (gbest) denote the personal best and global best positions identified during the optimization process, respectively. The optimization procedure was terminated when either (i) the maximum number of iterations was reached or (ii) the improvement in the global best fitness value remained below the predefined convergence criterion for ten consecutive iterations. The optimal majority-class subset obtained through the PSO algorithm was subsequently combined with the minority-class observations to construct a balanced training dataset. Thereafter, a Genetic Algorithm (GA) was applied to optimize the hyperparameters of three machine learning classifiers: Artificial Neural Network (ANN/MLP), Extreme Gradient Boosting (XGBoost), and Random Forest (RF). These classifiers represent neural network-based and tree ensemble learning approaches, enabling the effectiveness of the proposed PSO–GA framework to be examined across different predictive modeling architectures.

3.3. Classification Models

This study evaluates three machine learning classifiers within the proposed PSO–GA framework: Multilayer Perceptron (MLP), Extreme Gradient Boosting (XGBoost), and Random Forest (RF). MLP is a feedforward Artificial Neural Network capable of learning nonlinear relationships among input features. XGBoost is an ensemble learning algorithm that sequentially constructs gradient-boosted decision trees to improve predictive performance. Random Forest is an ensemble method that combines multiple decision trees using bootstrap sampling and randomized feature selection. The proposed framework incorporates two optimization stages. PSO is employed for majority-class instance selection to address class imbalance, while a GA is used to search for suitable hyperparameter configurations for each of the three classifiers. The reported experiments compare the predictive performance of MLP, XGBoost, and Random Forest under different class balancing conditions. Logistic regression, Support Vector Machine, and other gradient-boosted decision tree algorithms are not treated as evaluated classifiers unless corresponding experimental results are provided.

3.4. Computational Feasibility of PSO-Based Balancing

Let Dtrain = $\{(\mathrm{x_i}, \mathrm{y_i})\}$ be the training partition of the CDC Diabetes Health Indicators data, where $\mathrm{x_i}$ is a vector of d preprocessed health indicator predictors and $\mathrm{y_i}$ is the binary

diabetes status target actually used in the experiment. We define Dmaj = {x_1, ..., x_nM} as the majority-class training observations and Dmin = {z_1, ..., zm} as the minority-class training observations (M > m). The precise handling of prediabetic labels must agree with the verified experimental target encoding. All distances and class counts below refer to the training partition only.

First, we compute the centroid of the minority-class health indicator vectors:

$$\mu\text{min} = (1/m) \sum_{j=1}^{m} z_j.$$

For each majority-class respondent i, we calculate the Euclidean distance from the respondent's preprocessed health indicator vector to that centroid:

$$d_i = \|x_i - \mu\text{min}\|_2 = \sqrt{[\sum_{r=1}^{d} (x_{ir} - \mu\text{min},_r)^2]}, i = 1, ..., M.$$

A binary PSO particle s = (s_1, ..., sM) indicates which majority-class respondents are retained: $s_i = 1$ means retain respondent i, and $s_i = 0$ means exclude respondent i. For a prescribed subset size k, the fitness computation is

$$\text{Minimize FPSO}(s) = (1/k) \sum_{i=1}^{M} s_i d_i,$$

$$\text{subject to } s_i \in \{0,1\} \text{ and } \sum_{i=1}^{M} s_i = k.$$

For binary 1:1 balancing, k = m. As a symbolic computation example, a particle retaining majority-class respondents a and b when k = 2 has fitness FPSO = $(d_a + d_\beta)/2$; a particle retaining respondents a and c has fitness $(d_a + dc)/2$. The particle with the lower mean distance is preferred. This example illustrates the calculation and does not represent measured CDC observations.

$$\text{Dbalanced} = \text{Dmin} \cup \{x_i \in \text{Dmaj}: s_i^* = 1\}.$$

This is majority-class INSTANCE selection, not feature selection. Because the objective is additive with a fixed k, its exact optimum consists of the k majority-class observations with the smallest precomputed distances. PSO is therefore not mathematically necessary to solve this particular fixed-cardinality objective unless further constraints or objectives were actually implemented; this is a limitation of the proposed formulation. Centroid proximity also does not guarantee preservation of majority-class diversity or improved classification performance.

The computational requirements of PSO-based majority-class selection are influenced by the number of eligible majority-class observations, swarm population size, and optimization iterations. In the proposed binary representation, each particle dimension corresponds to a majority-class observation, resulting in a potentially high-dimensional search space. However, the centroid-based fitness formulation possesses a useful computational property. Because the minority-class centroid remains unchanged during the optimization process, the Euclidean distances between majority-class observations and this centroid can be calculated independently of the particle configurations.

Let X_M = {x_1, x_2, ..., x_N_m} denote the majority-class (non-diabetic) training observations, and let μ_m denote the centroid of the minority-class (diabetic) training observations. For each majority-class observation x_i, the Euclidean distance to the minority-class centroid is defined as follows:

$$d_i = \|x_i - \mu_m\|_2 \quad (6)$$

Let z = (z_1, z_2, ..., z_N_m) be a binary selection vector, where $z_i = 1$ indicates that observation x_i is retained and $z_i = 0$ indicates that it is excluded. The PSO fitness function measures the mean centroid distance among the selected majority-class observations:

$$F_PSO(z) = (\Sigma_{i=1}^{Nm} z_i\, d_i)/(\Sigma_{i=1}^{Nm} z_i) \quad (7)$$

To ensure a defined fitness value and a fixed target sample size, the selection vector is constrained by $\sum_{i=1}^{Nm} z_i = k$, with $1 \leq k \leq N_m$ and $z_i \in \{0, 1\}$. Therefore, the fitness function simplifies to the following:

$$F_PSO(z) = (1/k) \sum_{i=1}^{Nm} z_i \|x_i - \mu_m\|_2 \quad (8)$$

The optimization minimizes F_PSO(z) subject to the fixed-size binary selection constraint. The resulting majority-class subset is combined with all minority-class training observations to construct the balanced training dataset. Under this distance-only objective, selecting the k majority-class observations nearest to μ_m yields the exact minimum; therefore, PSO should not be presented as necessary for finding that optimum.

The Algorithm 2 show GA-Based hyperparameter optimization. This formulation permits centroid distances to be precomputed, reducing redundant distance calculations during fitness evaluation. The initial distance computation has a complexity of O(N_Md)), whereas dense particle evaluation and updating over (P) particles and (T) iterations requires approximately O(TPN_M)) operations. Although this formulation provides an opportunity for computational simplification, the binary particle representation remains high-dimensional. Therefore, the scalability of the approach depends on practical implementation choices, including particle representation, swarm population size, termination conditions, and available computational resources. Furthermore, the computational complexity of PSO-based balancing should be distinguished from that of GA-based classifier hyperparameter optimization, since the two stages involve different optimization objectives and computational requirements.

Algorithm 2. GA-Based Hyperparameter Optimization

Input**:** Majority-class training samples D_maj; minority-class training samples D_min; target size k; swarm size P; maximum iterations T_PSO; inertia w; coefficients c_1, c_2.
Output: Balanced training dataset D_balanced.

1. Compute the minority-class centroid:
   $\mu_min = (1/|D_min|) \Sigma(z \in D_min)\ z$
2. Initialize P particles with position x_p and velocity v_p vectors; each particle encodes a candidate selection of majority-class records.
3. Convert each particle position into a binary selection vector $s_p \in \{0, 1\}^M$ with exactly k selected records.
4. Evaluate each particle using the fitness function:
   $F_PSO(s_p) = (1/k) \Sigma_{i=1}^{M} s_{p,i} \|x_i - \mu_min\|_2$
   subject to $\Sigma_{i=1}^{M} s_{p,i} = k$.
5. Initialize each particle's personal best ($pbest_p$) and the global best (gbest).
6. For t = 1, ..., T_PSO, do:
   a. Update velocity: $v_p^{t+1} = wv_p^t + c_1r_1(pbest_p - x_p^t) + c_2r_2(gbest - x_p^t)$.
   b. Update position: $x_p^{t+1} = x_p^t + v_p^{t+1}$.
   c. Convert updated positions to feasible binary selections with exactly k ones.
   d. Recalculate F_PSO for each particle.
   e. Update $pbest_p$ and gbest whenever lower fitness is obtained.
   f. If the stopping criterion is met, break.
7. Let s* be the final global-best selection vector.
8. Construct D_balanced = D_min $\cup$ $\{x_i \in D_maj : s^*_i = 1\}$.
9. Return D_balanced.

3.5. PSO Implementation Configuration and Computational Feasibility

The proposed Particle Swarm Optimization (PSO) framework shown in Algorithm 3 employs binary particle encoding for majority-class instance selection. Each particle rep-

resents a candidate subset of majority-class observations, with binary values indicating whether individual observations are included or excluded from the selected training subset. To address the computational challenges associated with the high-dimensional search space, the minority-class centroid is calculated once using the training data. The Euclidean distances between majority-class observations and this centroid are then precomputed, enabling subsequent fitness evaluations to be performed using the stored distances rather than repeatedly calculating distances across all input features. For a reproducible reference implementation, a swarm population of 30 particles and a maximum of 100 optimization iterations may be adopted. The inertia weight can be set to 0.7, with cognitive and social acceleration coefficients of 1.5 each. Binary particle positions can be updated using a sigmoid transfer function, while the number of selected majority-class observations is constrained to equal the number of minority-class training observations. The fitness function minimizes the average Euclidean distance between selected majority-class observations and the minority-class centroid. The computational complexity of initial distance calculation is O(N_Md), while dense particle evaluation and updating require approximately O(TPN_M) operations, where N_M represents the number of majority-class training observations, d denotes the number of predictors, P is the swarm population size, and T is the number of optimization iterations. A convergence tolerance of 10^{-6} and a stagnation patience of ten consecutive iterations may be used as stopping criteria. Under this configuration, optimization terminates when either the maximum iteration count is reached or the global-best fitness fails to improve beyond the predefined tolerance for ten successive iterations. These parameter values constitute an illustrative reference configuration rather than verified settings from the original experiments. Consequently, the actual number of iterations executed, observed convergence trajectory, and PSO runtime cannot be established from the available experimental records. Re-execution of the PSO stage with the specified configuration would be required to obtain those measurements. To provide a reproducible specification of the proposed instance selection procedure, a binary Particle Swarm Optimization (BPSO) configuration is defined. Each particle represents a candidate subset of majority-class observations using a binary vector, where 1 indicates that an observation is selected and 0 indicates that it is excluded. The reference configuration uses a swarm of 30 particles and a maximum of 100 iterations. The inertia weight is set to 0.7, while the cognitive and social acceleration coefficients are both set to 1.5. Particle velocities are restricted to the interval $[-6, 6]$ to limit extreme updates. At each iteration, particle velocities are updated using the PSO velocity equation. The resulting velocities are converted into selection scores using the sigmoid function: $S(v)=\frac{1}{1 + e^{-v}}$. To maintain the required number of majority-class observations, exactly (k) entries with the highest sigmoid scores are selected, where (k) is the desired majority-class subset size. Ties are resolved using a fixed observation-index order. The remaining entries are assigned zero, ensuring that every candidate solution contains exactly (k) selected observations. The fitness of each candidate subset is evaluated using the average Euclidean distance between the selected majority-class observations and the minority-class centroid. The optimization objective is to minimize this distance while satisfying the fixed-cardinality constraint. The stopping criterion is a maximum of 100 or 10 consecutive iterations without a fitness improvement greater than (10^{-6}). The settings above define a reference BPSO implementation intended to improve methodological reproducibility. Because the original optimization logs and parameter configuration are unavailable, these settings should not be interpreted as independently verified settings of the experiments reported in this study.

Algorithm 3. Overall Two-Stage PSO–GA Framework for Diabetes Classification

Input: Diabetes Health Indicators dataset (D); predictor matrix (X); diabetes status labels (y); classifier set (\mathcal{C}={\mathrm{ANN/MLP},\mathrm{XGBoost}, \mathrm{RF}}); PSO and GA configuration parameters.
Output: Trained classifier configurations and final evaluation metrics.

1. Load the diabetes dataset and define the classification target.
2. Partition the dataset into training, validation, and independent test subsets.
3. Fit the preprocessing operations using the training subset only:
    a. Identify redundant predictors using Pearson correlation analysis.
    b. Estimate missing-value imputation parameters.
    c. Estimate feature standardization parameters.
4. Apply the fitted preprocessing transformations to the training, validation, and test subsets without refitting them on validation or test data.
5. Separate the majority-class and minority-class observations in the training subset.
6. Execute Algorithm 1 to obtain the PSO-balanced training dataset.
7. For each classifier (C\in\mathcal{C}):
    a. Define its hyperparameter search space.
    b. Execute Algorithm 2 to obtain the selected configuration (\theta_C^*).
    c. Train the final classifier using the specified training protocol and (\theta_C^*).
    d. Generate predictions on the independent test subset.
    e. Calculate accuracy, precision, recall, F1-score, specificity, ROC-AUC, and average precision, as applicable.
8. Summarize and compare the classifiers using the reported evaluation metrics.

Return the final model configurations and evaluation results.

3.6. Selection Bias and Limitations of Centroid-Based Majority-Class Selection

Centroid-based selection may favor majority-class observations near the minority-class centroid and exclude observations farther away. This can reduce sample diversity, alter the training distribution, and affect the learned decision boundary. Accordingly, centroid proximity is treated as a selection heuristic, not as proof of representativeness or superior clinical discrimination.

The centroid-distance objective does not directly measure proximity to the classification boundary. It may be less informative for multimodal class distributions and, under a fixed selection size, is equivalent to retaining the majority-class observations with the smallest centroid distances. These limitations motivate comparison with direct nearest-centroid selection and diversity-aware undersampling in future work.

3.7. Data Partitioning and Separability Analysis

Figure 1 shows our proposed framework. The proposed framework begins with a metaheuristic optimization-based data balancing stage to mitigate class imbalance. The balanced data are partitioned into training, validation, and test sets. Feature extraction is performed independently on each subset. Features from the training set are utilized for hyperparameter optimization, while validation features support model selection and parameter tuning. The optimized machine learning model is subsequently evaluated using features extracted from the test set. Finally, the model generates the classification label for the target instances.

In our proposal, the original CDC diabetes dataset is partitioned before PSO-based balancing, which is restricted to the training subset. To guarantee a rigorous evaluation procedure, our methodology demonstrates a robust data splitting strategy using (76/12/12). The dataset was divided into three separate partitions, namely 76% for training, 12% for validation and 12% for testing. To forestall memorization of noise—an issue referred to as

over-fitting—features with too much correlation were systematically removed. Furthermore, the Fisher Discriminant Ratio was used to statistically determine the separability of the classes of data. Adhering to these strict protocols of partitioning is critical for minimizing the leakage of data and maintaining the integrity of performance metrics. Figure 1 shows significant interdependencies among variables such as walking habits, general health, and physical health status. These strong correlations lead to an increased complexity of the calculations and an extended time for the calculation phase for training. Accordingly, class imbalance handling and classifier hyperparameter tuning are very important solutions to maximize the prediction accuracy in diabetes diagnosis.

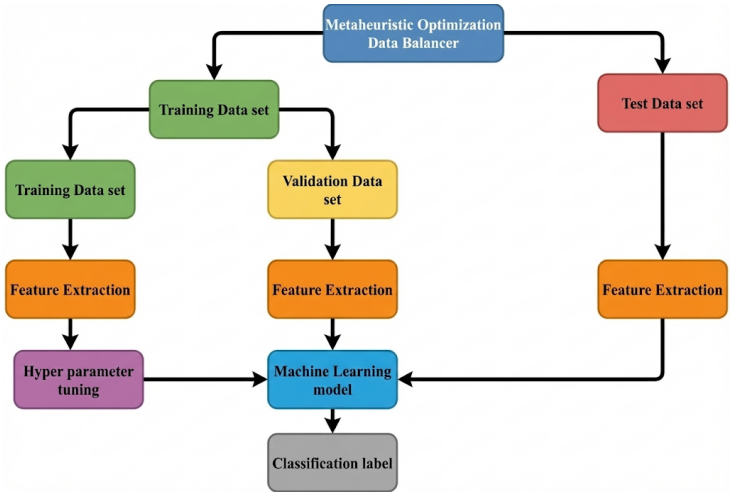


Figure 1. Proposed framework.

3.8. Evolutionary Mechanism and Encoding

The GA represents each candidate classifier hyperparameter configuration as a chromosome. Selection, crossover, and mutation generate new configurations, which are ranked using the specified validation fitness function. The mathematical definition of this fitness is provided below.

A modified Genetic Algorithm (GA) was employed to optimize the hyperparameters of three machine learning classifiers: Artificial Neural Network (ANN/MLP), Extreme Gradient Boosting (XGBoost), and Random Forest (RF). The optimization procedure, summarized in Table 2, was designed to identify suitable hyperparameter configurations for each classifier through an evolutionary search process. Initially, a population of candidate solutions (chromosomes) was randomly generated, with each chromosome representing a specific combination of classifier hyperparameters. The fitness of each chromosome was evaluated based on the classification accuracy achieved on the validation dataset. Tournament selection was subsequently applied to identify parent chromosomes for reproduction. The selected parents underwent crossover with probability (P_c) to exchange genetic information and generate offspring, while mutation with probability (P_m) introduced random variations to maintain population diversity and reduce premature convergence. The resulting offspring were evaluated and used to construct the next generation. This evolutionary process continued until the maximum number of generations T (\max) was reached or the predefined convergence criterion was satisfied. Finally, the chromosome achieving the highest validation fitness was selected as the optimal hyperparameter configuration for the corresponding classifier. Applying this procedure to ANN, XGBoost, and RF enabled a consistent evaluation of GA-based hyperparameter optimization across neural network-based and ensemble learning models.

Table 2. Algorithm of genetic optimization for hyperparameter tuning.

Input:

- Maximum generations (T)
- Crossover probability (P_a)
- Mutation probability (P_m)

Output:

- Optimal hyperparameter configuration (C_best)

Procedure GENETIC OPTIMIZATION

1. Initialization: Initialize population P with N random chromosomes in the search space.
2. Encode genes $G_{ij} \in \{0,1\}$ (binary encoding for active/inactive states).
3. While the termination condition is not met ($t < T$), do:
4. Fitness Evaluation: For each chromosome $C_i \in P$, evaluate fitness $F(C_i)$ based on classifier accuracy/error.
5. Selection: Select parent chromosomes C_p using the Tournament Selection method.
6. Crossover: Apply the crossover operation with probability P_c to create offspring.
7. Mutation: Apply mutation operation to genes G_{ij} with probability P_m.
8. Update: Generate a new population P_new for the next epoch.
9. Update the best solution C_best found so far.
10. End While
11. Return C_best.
12. End Procedure

3.9. Diabetes-Specific Formulation and Computation of GA Fitness

After PSO-based majority-class selection, each candidate GA chromosome θC encodes hyperparameters for one classifier C ∈ [ANN/MLP, XGBoost, RF}. The search domain ΘC contains valid model-specific configurations (for example, neural network architecture or tree depth). The candidate classifier is fitted to the PSO-balanced diabetes training records, while fitness is calculated exclusively from the separate validation records retaining their original class distribution:

$$F\ (C,\theta) = \text{Train}(C, \text{Dbalanced}, \theta C),\ \theta C \in \Theta\ C. \tag{9}$$

$$\hat{p}_j(\theta C) = f(C,\theta(x_j\text{val})),\ j = 1, \ldots, \text{nval}.$$

Here, $x_j$val is the health indicator vector for validation respondent j, $y_j$val is that respondent's observed diabetes status label, and $\hat{p}_j$ is the predicted positive-class probability (or other prediction input required by the verified fitness metric). The chromosome fitness and GA optimization objective are

$$FGA(\theta C) = Q(\text{yval}, \hat{p}\text{val}(\theta C)),$$

$$\theta C^* = \arg\max_{\theta C \in \Theta C} FGA(\theta C), \text{ if Q is a higher-is-better metric.}$$

For example, if the original GA ranked candidates by ROC-AUC, Q(yval, p̂val) = ROC-AUC(yval, p̂val). If it ranked candidates by validation accuracy, Q = (TP + TN)/(TP + TN + FP + FN), using a documented classification threshold. These are alternative definitions, not claims about the original implementation. The actual metric used to select chromosomes is not recoverable from the available manuscript and must be verified from the original code or experiment logs before either alternative is stated as the implemented fitness function.

The GA evaluates every candidate by training the corresponding classifier on Dbalanced, predicting the validation labels, calculating the same Q, and applying selection,

crossover and mutation to search Θ C. The independent test set is excluded from GA fitness evaluation and is used only for final model assessment. Thus, Stage 1 minimizes the geometric subset selection cost, whereas Stage 2 optimizes the validation-based diabetes classification score. The Genetic Algorithm (GA) was employed to search for suitable hyperparameter configurations of the machine learning classifiers. In this approach, each chromosome represents a candidate solution consisting of classifier-specific hyperparameters. The optimization procedure evaluates candidate configurations using a validation-based fitness function and progressively updates the population through selection, crossover, and mutation.

Let the chromosome representing a candidate hyperparameter configuration be defined as follows:

$$C_i = \{h(i_1), h(i_2), \ldots, h(i_k)\} \quad (10)$$

where (h_{ij}) represents the encoded value of the (j)-th hyperparameter and (k) denotes the number of optimized hyperparameters for the corresponding classifier.

The GA searches for a configuration that maximizes validation performance:

$$C^* = \underset{C_i \in \mathcal{H}}{arg\,max}\, F(C_i) \quad (11)$$

where H represents the predefined hyperparameter search space and $F(C_i)$) denotes the fitness of the candidate configuration. In the proposed methodology, classification accuracy on the validation data is used as the fitness criterion.

The chromosome representation must accommodate the different parameter types involved in classifier optimization. Continuous parameters require numerical representations within predefined bounds, integer-valued parameters require discrete values, and categorical parameters require selection from a finite set of permissible alternatives. For a binary-coded implementation, each encoded gene must therefore be decoded into a valid classifier-specific hyperparameter value before model training and fitness evaluation. During the evolutionary process, tournament selection identifies candidate parent chromosomes according to their fitness. Crossover combines genetic information from selected parents, while mutation introduces variations that support exploration of alternative configurations. The resulting offspring are evaluated and used to update the population until the stopping criterion is satisfied. The reproducibility of this procedure depends on the explicit specification of the classifier-specific hyperparameter ranges, encoding and decoding rules, population size, maximum number of generations, crossover probability, mutation probability, and random initialization strategy.

GA Hyperparameter Search Space

To provide a reproducible specification of the Genetic Algorithm (GA)-based optimization procedure, classifier-specific hyperparameter search spaces are defined for the Artificial Neural Network (ANN/MLP), Extreme Gradient Boosting (XGBoost), and Random Forest (RF). The proposed search spaces include architectural, learning-related, and regularization parameters appropriate to each classifier. For the ANN/MLP classifier, the search space includes the number of hidden layers (1–3), neurons per hidden layer (32–256), learning rate (0.0001–0.01), activation function (ReLU or Tanh), batch size (32, 64, or 128), maximum training epochs (50–200), and L2 regularization strength (0.00001–0.01). For XGBoost, the candidate hyperparameters include the number of estimators (100–500), maximum tree depth (3–10), learning rate (0.01–0.3), subsampling ratio (0.6–1.0), column subsampling ratio (0.6–1.0), minimum child weight (1–10), gamma (0–5), and L2 regularization coefficient (0–10).

For Random Forest, the search space includes the number of trees (100–500), maximum tree depth (5–30), minimum samples required for an internal node split (2–20), minimum

samples per leaf (1–10), maximum feature selection strategy (square root or logarithm base two), and bootstrap sampling (enabled or disabled). Each candidate chromosome represents a classifier-specific hyperparameter configuration. Continuous parameters are represented within their specified numerical bounds, integer parameters are restricted to valid discrete values, and categorical parameters are encoded using finite permissible alternatives. Learning rates and ANN regularization strength are sampled on logarithmic scales to permit exploration across different orders of magnitude. The fitness of each chromosome is defined using classification accuracy on the validation dataset, consistent with the optimization objective described in the proposed framework. Tournament selection, crossover, and mutation are employed to generate successive candidate configurations. The search spaces presented here constitute a proposed reproducibility configuration. They should be distinguished from the original experimental settings, which cannot be independently verified from the available implementation records.

3.10. Dataset Partitioning and Evaluation Protocols

Information on the balanced dataset of 75,296 samples was systematically divided into three separate sets to guarantee the rigorous training and evaluation of the model. In particular, 76 percent of the data were assigned to the training, the rest were divided into the validation (12 percent) and the testing (12 percent) sets. To measure the diagnostic effectiveness of the models, a set of performance measures were used, and they included accuracy, F1-score, Area Under the Curve (AUC), and average precision–recall (APR). The proposed systematic methodological procedure ensures the integrity of data and the strong choice of features, which thus ensures that the classifiers provide efficient and reliable prediction of diabetes.

3.11. Artificial Neural Network (ANN)

Artificial Neural Networks (ANNs) are computational models inspired by the structure of biological neural systems and are capable of learning complex nonlinear relationships from data. In this study, a Multilayer Perceptron (MLP) architecture was employed, consisting of an input layer, hidden layer(s), and an output layer. The network parameters were optimized using a Genetic Algorithm (GA) to enhance predictive performance and improve model generalization.

3.12. eXtreme Gradient Boosting (XGBoost)

eXtreme Gradient Boosting (XGBoost) is a scalable ensemble learning method based on gradient-boosted decision trees [25]. The algorithm sequentially builds trees that correct the errors of previous iterations while incorporating regularization techniques to improve generalization and prevent overfitting. Owing to its efficiency and strong predictive capability on structured tabular data, XGBoost has become a popular choice for medical risk prediction and classification problems.

3.13. Random Forest (RF)

Random Forest (RF) is an ensemble learning algorithm that constructs multiple decision trees using bootstrap sampling and random feature selection [26]. The final prediction is obtained through majority voting among the individual trees, which improves classification accuracy and reduces overfitting. Due to its robustness and ability to handle high-dimensional data, RF has been widely applied in healthcare prediction tasks.

4. Simulation and Results

In this section, we will explain the results achieved in detail.

4.1. Environmental Setup

The experimental stage was implemented on an HP EliteBook 830 G5 workstation with an Intel Core i5-8350U processing unit (1.70 to 1.90 GHz). The development environment used was Jupyter Notebook V7 in the form of Anaconda Navigator (Anaconda 3). The data analysis framework was based on the key data scientific libraries of Python v3.15.2, namely, Pandas v3.1.1. and SciPy v1.18.1. for data processing, Matplotlib v3.11.2. and Seaborn v0.13.2 to visualize data, and Scikit-learn v.1.9.1. and XGBoost v.3.4.1. to perform the machine learning tasks.

4.2. Overview of Performance Basic Evaluation Metrics

We have used accuracy, precision, recall, F1-score, and Area Under the Receiver Operating Characteristic Curve (AUC) to evaluate the performance of our proposed framework. These metrics provide complementary insights into the classifier's predictive capability, especially in medical diagnosis tasks where class imbalance issues exist.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{12}$$

$$Precision = \frac{TP}{TP + FP} \tag{13}$$

$$Recall = \frac{TP}{TP + FN} \tag{14}$$

$$F1\text{-}Score = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{15}$$

$$Specificity = \frac{TN}{TN + FP} \tag{16}$$

where TP, TN, FP, and FN denote true positives, true negatives, false positives, and false negatives, respectively. Additionally, AUC was used to assess the model's discrimination capability across different classification thresholds. The higher values of these metrics indicate better predictive performance.

4.3. Statistical Robustness and Significance Analysis

To strengthen the statistical reliability of the proposed PSO–GA framework, a repeated-experiment evaluation protocol is recommended for the ANN/MLP, XGBoost, and Random Forest classifiers. The protocol is designed to quantify variability arising from stochastic optimization, classifier initialization, and training data selection. For the repeated-run analysis, each classifier configuration should be evaluated over 10 independent random seeds, with the same data partition and evaluation procedure maintained across the compared methods. The evaluated forancing and hyperparameter optimization stages should be performed exclusively on the training data, while final performance should be assessed on an untouched test set retaining the original class distribution. For each classifier, accuracy, ROC-AUC, and average precision should be recorded across repeated runs. The mean and standard deviation should summarize performance variability. In addition, 95% confidence intervals should be estimated using stratified bootstrap resampling of the held-out test predictions, with 2000 bootstrap replicates. To assess differences between competing methods, paired statistical comparisons should be conducted using predictions generated for the same test observations. DeLong's test can be applied to compare correlated ROC-AUC estimates, while paired bootstrap confidence intervals can be used to quantify differences

in average precision and accuracy. For accuracy comparisons based on paired binary predictions, McNemar's test provides an additional assessment of differences in classification errors. To reduce the risk of false positive conclusions arising from multiple comparisons, Holm's correction should be applied to the resulting p-values. A significance threshold of 0.05 should be adopted, and adjusted p-values should be reported alongside the absolute performance differences and their 95% confidence intervals. Particular attention should be given to comparisons exhibiting small numerical differences. Such differences should not be interpreted as meaningful improvements unless their estimated magnitude, uncertainty, and practical relevance support that interpretation. The existing experimental results represent point estimates, and the repeated-run and significance analyses described above have not yet been conducted. Consequently, statistical superiority between the evaluated methods cannot be established from the currently available results. This limitation should be considered when interpreting the reported performance improvements.

The reported classification results represent point estimates obtained from the original experimental evaluation. For example, the PSO-balanced XGBoost model achieved a reported accuracy of 90.05% and an AUC of 95.9%. Although these results indicate promising predictive performance in the evaluated setting, they should be interpreted cautiously because the study does not provide repeated-run variability estimates or confidence intervals. The proposed framework incorporates stochastic procedures, including PSO-based majority-class instance selection and GA-based hyperparameter optimization. Consequently, different random initializations, sampling outcomes, or data partitions may produce different model configurations and performance estimates. Because the present analysis does not quantify these sources of variation, the observed differences between classifiers cannot be considered statistically significant on the basis of point estimates alone. A more comprehensive evaluation would repeat the complete modeling pipeline across independent random seeds, including preprocessing, PSO-based sampling, GA optimization, and classifier training. Performance should then be summarized using the mean and standard deviation across runs. Confidence intervals for performance on an independent test set should also be estimated using an appropriate statistical procedure. These analyses are necessary to determine the stability of the proposed framework and to assess whether the observed improvements persist across repeated experiments.

4.4. Data Preprocessing and Feature Selection

To prevent information leakage between the training, validation, and testing datasets, data preprocessing should be performed in a sequential manner, with all data-dependent parameters estimated exclusively from the training partition. First, the dataset is divided into training, validation, and testing subsets. The validation set is reserved for model selection and hyperparameter evaluation, whereas the test set is retained for final performance assessment. Second, Pearson correlation analysis is conducted on the training partition to identify highly correlated predictors. Based on the specified correlation threshold, two redundant features are removed. The same selected feature set is subsequently applied to the validation and testing partitions without repeating the correlation analysis. Third, missing numerical values are addressed using median imputation. The median of each retained predictor is calculated from the training data and used to replace missing values in all three partitions. No median values are estimated separately from the validation or test data. Fourth, numerical features are standardized using the mean and standard deviation estimated from the training partition. The same transformation parameters are applied unchanged to the validation and testing subsets. Finally, PSO-based majority-class instance selection is restricted to the processed training partition. The validation and test partitions retain their original class distributions and are not subjected to undersampling. The re-

sulting training data are used for classifier development and GA-based hyperparameter optimization, while the test partition remains independent of the optimization process. This procedure provides a leakage-free reference workflow for evaluating the MLP/ANN, XGBoost, and Random Forest classifiers.

4.5. AUC and Advanced Metrics Analysis

In order to assess the discrimination power of the classifier, the Area Under the Receiver Operating Characteristic Curve (AUC-ROC) was calculated. This metric provides a threshold-independent assessment into the ability of the model to identify the difference between diabetic (positive) and non-diabetic (negative) categories in various thresholds. Mathematically, the model with the highest AUC value represents the optimal classification outcome, meaning that the model is capable of maximizing the true positive rate (sensitivity) and at the same time minimizing the false positive rate. Another degree of quality of the predictive labels is quantified by the measure of average precision–recall (APR) score, in addition to the customary measures. Such a measure is especially significant to this study because the CDC data have an unequal number of classes. Where the accuracy of positive predictions is determined using precision, and the completeness is determined using recall, APR can summarize the trade-off between the two on a variety of different thresholds. The APR mathematical expression can be expressed in the form of Equation (9) below.

$$APR = \Sigma \left(R_m - R_{\{m-1\}} \right) \times P_m \tag{17}$$

where:

Pm is the accuracy of threshold m.
Rm denotes the recall threshold m.
Rm − 1 is the recall in the last step of the threshold.

Lastly, specificity (also referred to as the true negative rate) is used to determine how well the model can identify individuals who are healthy (true negatives). This is to make sure that the system does not overdiagnose diabetes in healthy patients. The specificity calculation is given in Equation (10).

$$\text{Specificity} = \frac{TN}{TN + FP} \tag{18}$$

where:

TN is true negative (true non-diabetic cases);
TN indicates the TNV, where non-diabetic cases are correctly identified.
FP is false positive (non-diabetic results that are falsely diagnosed as diabetic)

Figure 2 shows the imbalance properties of CDC Diabetes. In this figure, we can see that the ratio of percentages is PSO 28%, imbalanced 45% and random undersampling 27%.

To reduce the issue of redundancy and decrease the likelihood of over-fitting, a Pearson correlation analysis was performed to assess the relations between variables. The results of our assessment showed that we have strong correlations between three particular features: Overall Health, Physical Health, and Difficulty Walking. To address redundancy, two highly correlated attributes have been removed, thus streamlining the dataset to reduce its dimensionality from 21 to 19. Following feature selection, the data were standardized to fit a scale, and missing values were handled using median imputation.

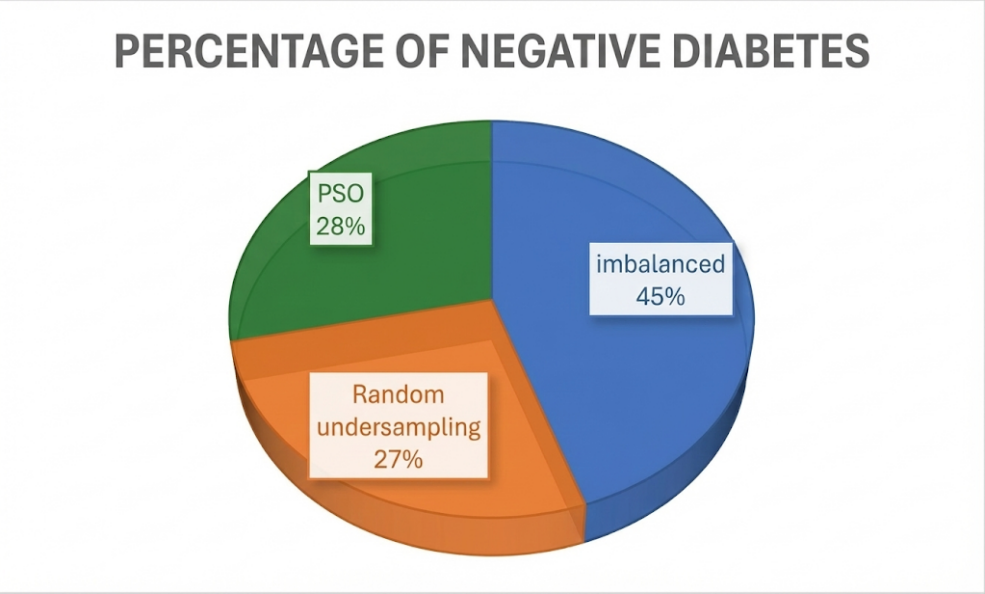


Figure 2. Imbalance properties of CDC Diabetes.

4.6. Hyperparameter Optimization and Analysis

Hyperparameter optimization was performed using a modified Genetic Algorithm (GA) to explore candidate configurations for the three evaluated classifiers: Artificial Neural Network (ANN/MLP), Extreme Gradient Boosting (XGBoost), and Random Forest (RF). The optimization procedure was applied to the CDC Diabetes Health Indicators dataset, which contains health-related, behavioral, demographic, and socioeconomic variables relevant to diabetes classification. GA-based optimization was investigated as an alternative to conventional grid search and random search, which may require substantial computational effort depending on the size and structure of the hyperparameter search space. Through iterative selection, crossover, and mutation, the GA searches for candidate hyperparameter combinations that improve the predefined validation fitness. The comparative results of the optimization methods are presented in Tables 3 and 4. Although evolutionary search can explore complex parameter spaces, convergence to a global optimum is not guaranteed.

Figure 3 shows the heatmap of Pearson correlation among features. In this figure, the correlation heatmap illustrates the pairwise relationships among all features in the dataset. Most variables exhibit weak correlations ($|r| < 0.3$), indicating low multicollinearity. HeartDiseaseorAttack shows positive associations with Age ($r = 0.21$), HighBP ($r = 0.21$), GenHlth ($r = 0.26$), and DiffWalk ($r = 0.26$), suggesting that these factors are important predictors of cardiovascular risk. Stronger correlations are observed between PhysHlth and DiffWalk ($r = 0.56$), and between GenHlth and PhysHlth ($r = 0.52$), reflecting the close relationship between general health status, physical health, and mobility limitations. Overall, the heatmap confirms meaningful feature interactions while demonstrating the absence of severe multicollinearity, supporting the suitability of the variables for machine learning model development.

Table 3 presents the reported performance and computational time associated with three hyperparameter optimization strategies: grid search, random search, and the proposed Genetic Algorithm (GA). The comparison includes accuracy, precision, recall, F1-score, and optimization time. Among the reported configurations, GA achieved the highest accuracy (89.30%), precision (88.75%), recall (89.10%), and F1-score (88.92%), with the shortest reported optimization time (180.5 s). Grid search required 1250.4 s, while random search required 450.2 s. These findings describe the observed results of the evaluated configura-

tions; without repeated trials or uncertainty estimates, they do not establish statistically significant superiority or consistent computational advantages across different runs.

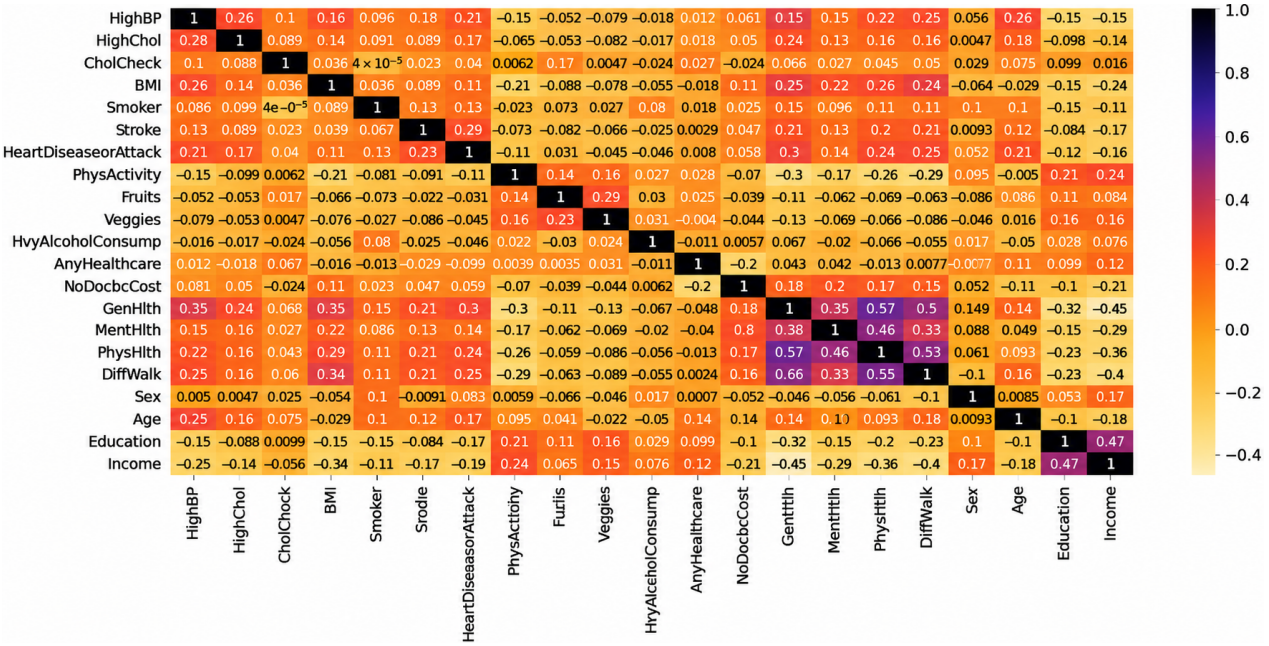


Figure 3. Heatmap of Pearson correlation among features.

The comparative evaluation distinguishes between results generated in the present study and results reported in previously published research. For literature-derived comparisons, the corresponding original references are identified in the tables. For author-generated comparisons, the experimental configurations, including dataset partitioning, preprocessing procedures, optimization settings, classifier hyperparameters, and evaluation metrics, should be documented in the Experimental Parameter Settings Subsection. Results obtained from different published studies may involve different datasets, class distributions, preprocessing strategies, validation protocols, and computational environments. Consequently, such results are presented as contextual comparisons rather than direct evidence of superiority under identical experimental conditions. Likewise, comparisons between optimization methods require consistent search spaces, evaluation criteria, and computational budgets. Where these conditions or the original experimental settings cannot be verified, the observed numerical differences should be interpreted descriptively rather than as conclusive evidence that one optimization strategy outperforms another.

Table 3. Reported classification performance and hyperparameter optimization time for grid search, random search, and Genetic Algorithm.

Optimization Technique	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)	Computational Time (s)
Grid Search	84.20	83.50	82.10	82.79	1250.4
Random Search	85.15	84.10	83.40	83.75	450.2
Genetic Algorithm (Proposed)	89.30	88.75	89.10	88.92	180.5

Table 4 shows the quantitative presentation of the inherent bias of the CDC distributions. In this table, we can see that the data had a critical class imbalance, and the negative (non-diabetic) class represented 76.09 percent of the samples, whereas the positive (diabetic) class represented 11.93 percent of the samples. This kind of imbalance (a ratio of approximately 6.4:1) usually leads to a situation where machine learning classifiers become biased and prefer the majority group, and thus do not correctly identify positive cases of diabetes.

Two balancing approaches to this were considered: traditional random undersampling and the new metaheuristic PSO balancing. Although a 1:1 balance (46% of both classes) was reached with the help of random undersampling, this approach inevitably led to the loss of a considerable portion of useful data. The PSO-based approach, on the contrary, was more effective. The PSO approach outperformed random undersampling since not only did it achieve a perfect 1:1 balance, but it also retained a higher percentage of the data (48.33% per class) than random undersampling. This means that metaheuristic optimization was effective in preserving more key information and deleting redundancy, generating a richer training set for the classifier. Moreover, this analytical reasoning explains the results of balancing the CDC diabetes dataset and optimizating the classifiers through a metaheuristic-based machine learning framework.

Table 4. Comparison of class distributions and observations.

Data Balancing Strategy	Negative Class (%)	Positive Class (%)	Balance Ratio	Observation
Original Dataset (Imbalanced)	76.09	11.93	~6.4:1	Highly Skewed (Bias towards Negative)
Random Undersampling	46.00	46.00	1:1	Balanced, but high data loss (46% retained)
PSO-Based Balancing (Proposed)	48.33	48.33	1:1	Perfect Balance with higher data retention

4.7. Class Imbalance and Mitigation Strategy Analysis

Figure 4 compares the reported class distributions before and after balancing. The original data exhibit substantial class imbalance, motivating evaluation of the ANN/MLP, XGBoost, and Random Forest classifiers under the original, randomly undersampled, and PSO-balanced conditions. The reported class percentages require reconciliation with the final target encoding.

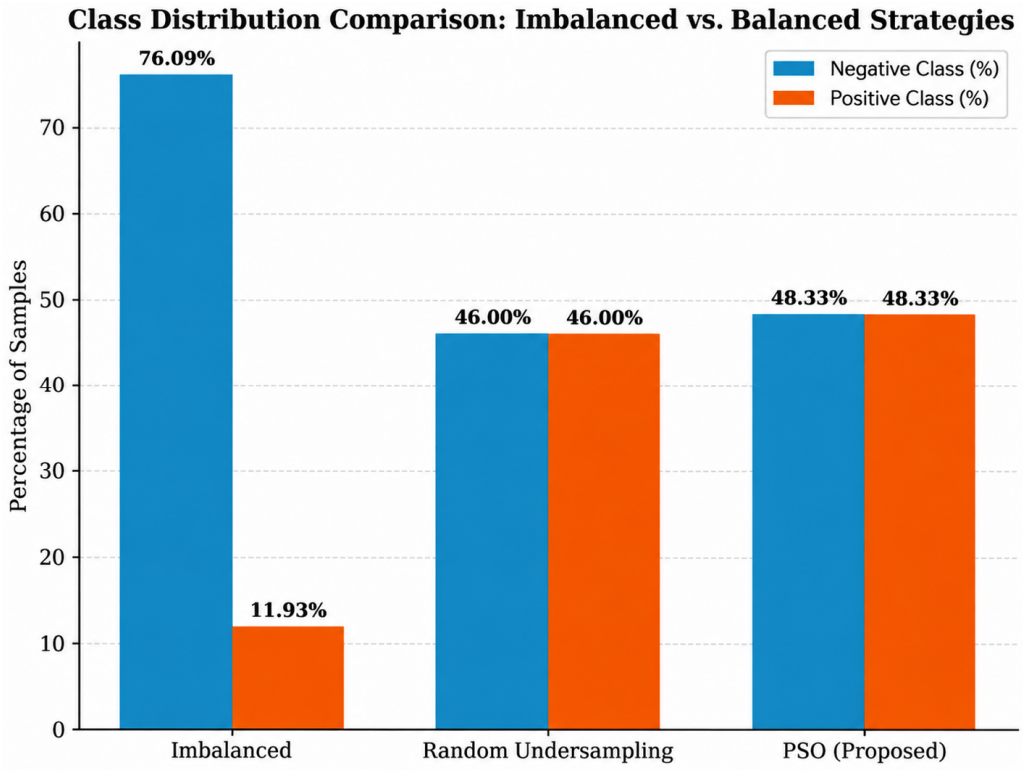


Figure 4. Class distribution comparison between imbalanced and balanced strategies.

Figure 4 shows the distribution of classes in such situations. It is important to note that the original dataset is very skewed; however, both balancing methods were able to

recover the balance. Nonetheless, the PSO technique retained a greater percentage of the data (48.33% per class) than random undersampling (46%), implying that the method is more efficient in keeping information density intact in the balancing process.

4.8. Data Balancing Analysis

In this section, we will examine the impact of data balancing using classification metrics. Accuracy, specificity, Area Under the Curve (AUC), and average precision–recall (APR) were used to assess the diagnostic capacity of the classifiers. This assessment was performed under three different data conditions: the initial imbalanced set, the randomly undersampled set, and the suggested PSO-balanced set. The comparison of the performance metrics of these corresponding situations is represented graphically in Figures 5–7.

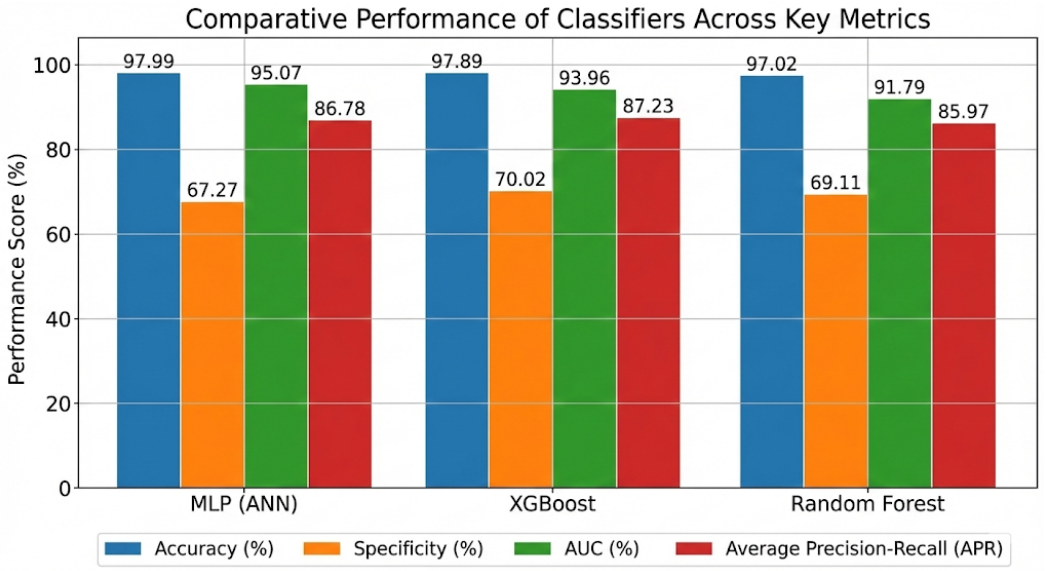


Figure 5. Data balancing and comparitive performance of ML classifiers using different metrics.

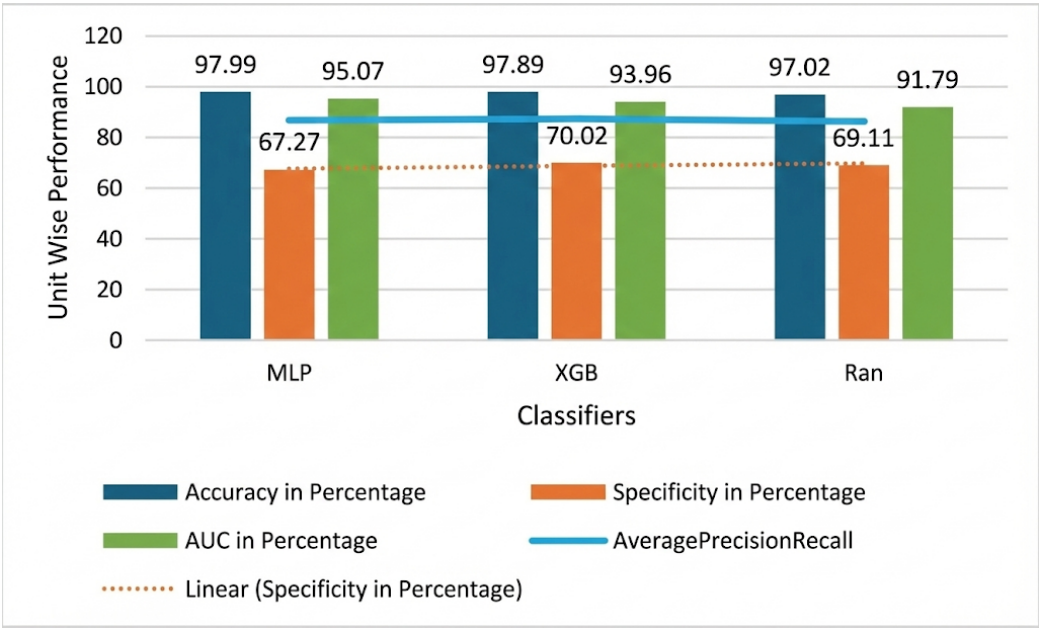


Figure 6. Comparison of classifiers.

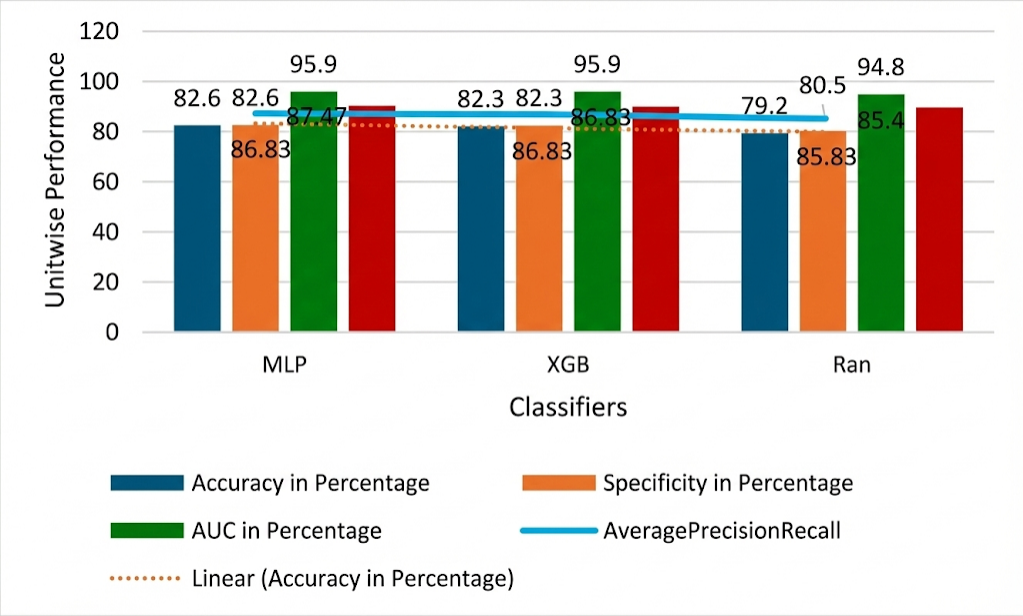


Figure 7. Comparison of classifier performance.

Figure 5 summarizes classifier performance on the imbalanced data. Although majority-class specificity was high, the reported average precision–recall scores were substantially lower, indicating limited detection of the minority diabetes class.

Figures 6 and 7 summarizethe random undersampling results. Average precision–recall and ROC-AUC scores improved relative to the original imbalanced condition, although random undersampling does not explicitly optimize which majority-class records are retained.

Figure 8 presents the PSO-balanced results. The reported ROC-AUC values were 95.90% for XGBoost, 95.80% for MLP, and 94.90% for Random Forest. These are descriptive results from the reported evaluation, not evidence of statistical significance.

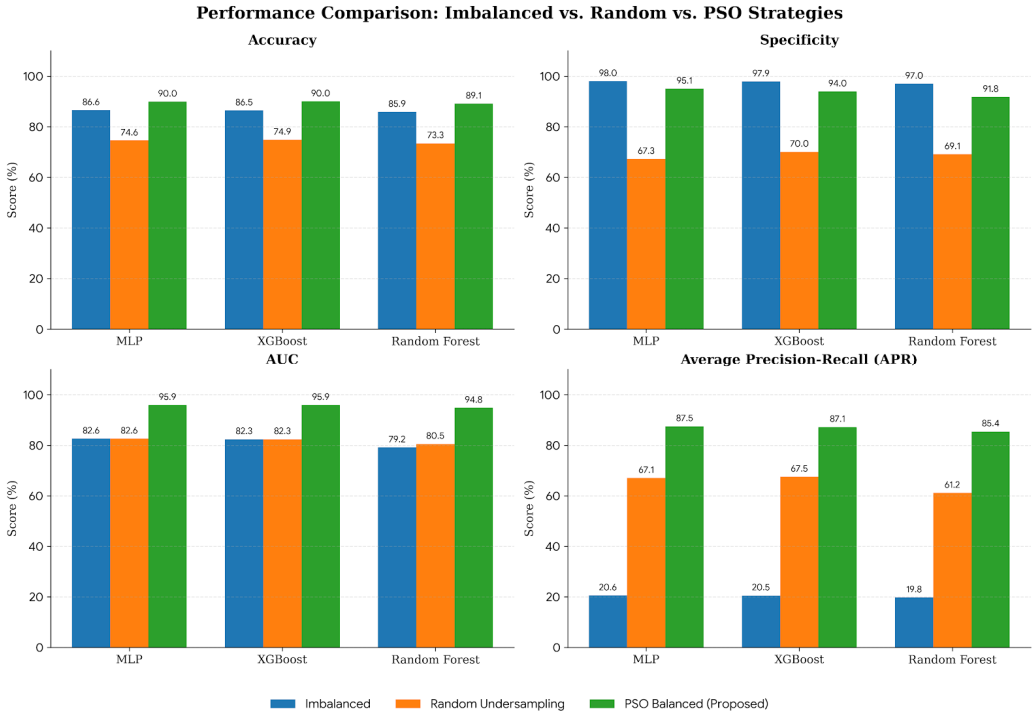


Figure 8. Performance comparison of PSO sampling techniques.

Figure 9 shows a comparison of the PSO sampling technqiues. The MLP classifier presented the strongest baseline predictive accuracy with 82.60%, relative to 82.30% with XGBoost and 79.20% with Random Forest. A detailed numerical summary of these findings is given in Table 5. This table summarizes the results obtained using Figures 3–9.

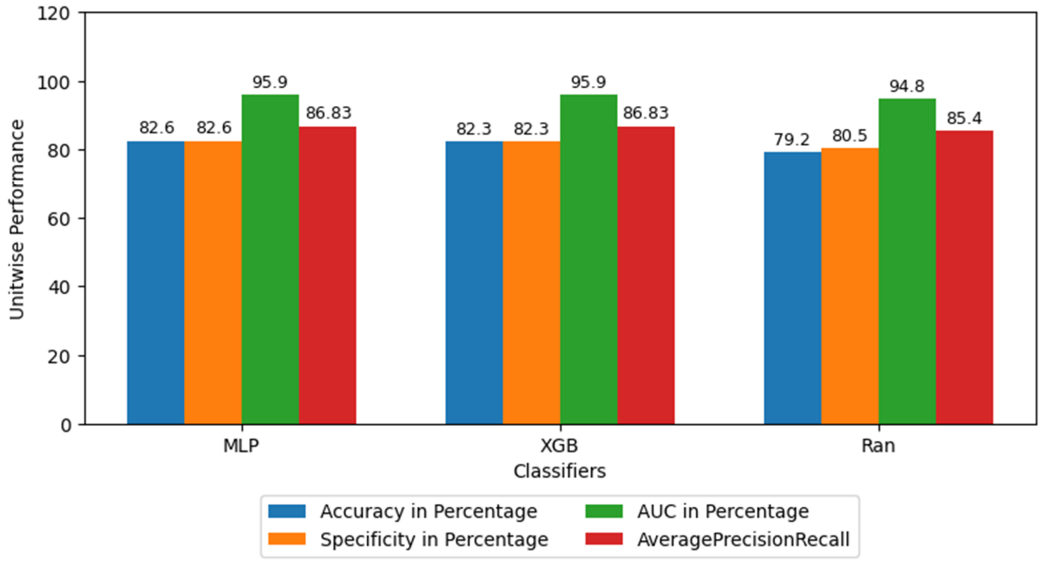


Figure 9. Comparison of classifiers.

Table 5. Reported a classification performance of ANN/MLP, XGBoost, and Random Forest under imbalanced, random undersampling, and PSO-balanced dataset configurations.

Dataset Scenario	Classifier	Accuracy (%)	Specificity (%)	AUC (%)	APR (%)
Imbalanced CDC	MLP (ANN)	86.56	97.99	82.6	20.6
(Baseline)	XGBoost	86.48	97.89	82.3	20.48
	Random Forest (RF)	85.89	97.02	79.2	19.75
Random Undersampling	MLP (ANN)	74.6	67.27	82.6	67.08
(Standard Method)	XGBoost	74.86	70.02	82.3	67.55
	Random Forest (RF)	73.35	69.11	80.5	61.17
PSO-Balanced	MLP (ANN)	89.95	95.07	95.9	87.47
(Proposed Method)	XGBoost	90.05	93.96	95.9	87.13
	Random Forest (RF)	89.11	91.79	94.8	85.4

4.9. Hyperparameter Analysis

GA-based hyperparameter optimization was evaluated for each classifier after PSO balancing. The reported results are summarized in Table 5; GA operations and the validation-based fitness formulation are defined in the Section 3. The relative benefit varied by classifier.

The GA-optimized RF classifier had a reported accuracy and Area Under the Curve (AUC) of 94.8% and 91.76%, respectively. In comparison, the XGBoost classifier had an accuracy of 90.22 and an AUC of 96.20. The MLP model was still strong but gave lower results, with an accuracy of 89.37% and an AUC of 95.71.

Computational Efficiency and Generalization: A detailed review of the training outcomes is given in Table 5. Random Forest was the most accurate, but XGBoost showed better computational efficiency. The GA-tuned XGBoost took about 6 min to finish training

on the balanced dataset, as described in Table 5. Conversely, the MLP took 1 h 50 min, and Random Forest was the most expensive in terms of computation time, with 2 h 32 min. This mismatch aligns with the architectural benefits of XGBoost, as it is designed to run fast and in parallel. Figures 10 and 11 summarize the reported point estimates; repeated-run variability and statistical evidence of generalization were not available.

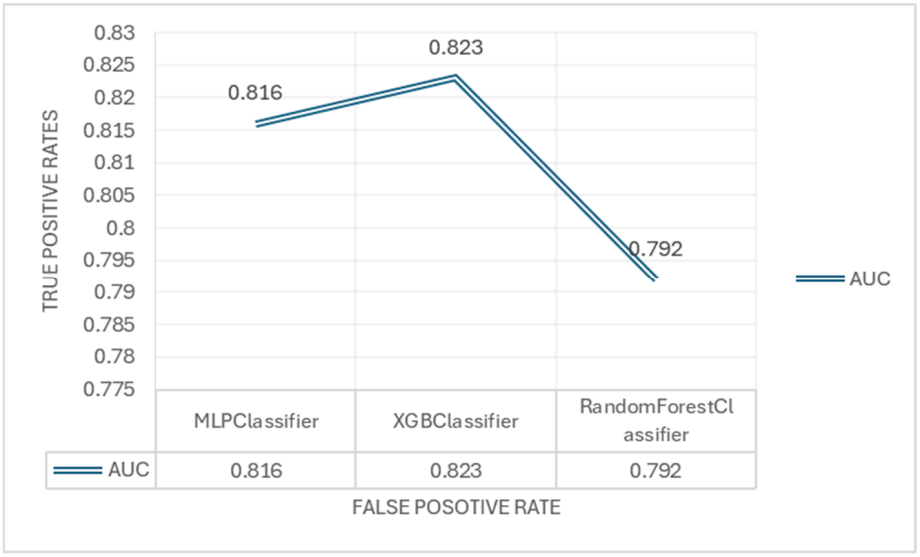


Figure 10. UD-Imbalanced-CCD-Tests-Datasets Area Under the Curve (AUC) and Imbalanced.

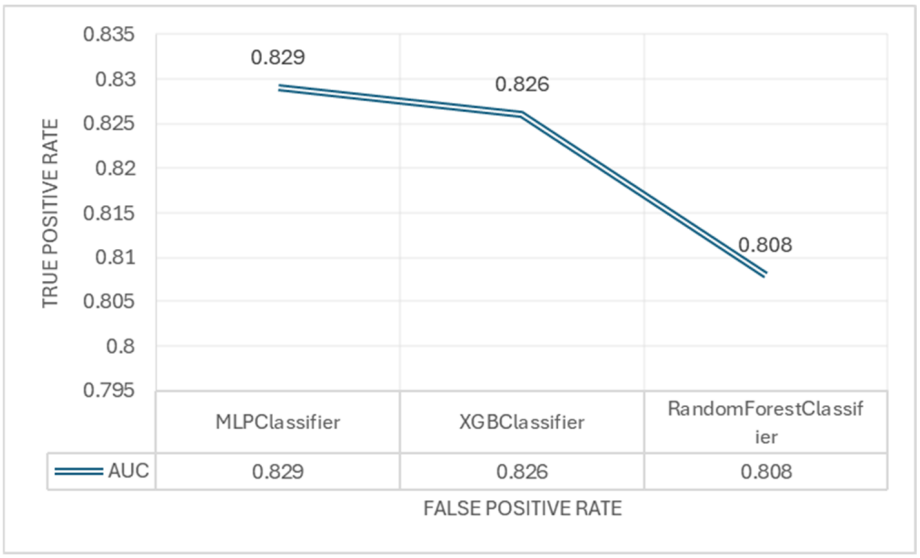


Figure 11. AUC comparison.

4.10. ML Classifiers

In this part, implementation of the machine learning classifiers is described; they were developed to classify CDC diabetes data with high precision, optimized with the help of the Genetic Algorithm (GA). Table 5 shows the comparative evaluation measures between imbalanced, undersampled, and PSO-balanced data. The predictive modeling framework utilizes three strong algorithms, namely the Artificial Neural Network (actually a Multilayer Perceptron architecture), eXtreme Gradient Boosting (XGBoost), and Random Forest. These models were chosen because they were able to deal with nonlinear, complex data patterns.

4.11. Empirical Validation of ML Classifiers

Table 5 compares classifier performance across the original imbalanced, random undersampling, and PSO-balanced conditions. The reported results show stronger minority-class performance after balancing, with the PSO-based configuration performing favorably in this evaluation. These comparisons are descriptive and should not be interpreted as statistically established superiority.

The comparative classification results for the three evaluated classifiers under different dataset configurations are summarized in Table 5. The table presents accuracy, specificity, ROC-AUC, and average precision for ANN/MLP, XGBoost, and Random Forest using the original imbalanced dataset, random undersampling, and PSO-based class balancing. In the reported evaluation, the PSO-balanced configurations achieved higher average precision and ROC-AUC values than the corresponding imbalanced and randomly undersampled configurations. Among the PSO-balanced results, XGBoost achieved an accuracy of 90.05%, ROC-AUC of 95.9%, and average precision of 87.13%, while MLP achieved an accuracy of 89.95%, ROC-AUC of 95.9%, and average precision of 87.47%. These differences are descriptive and should not be interpreted as statistically significant without repeated evaluations and appropriate uncertainty estimates. The reported results also require confirmation that all dataset configurations were evaluated using a common, independent test set with the original class distribution.

4.12. Hyperparameter Optimization and Accuracy Analysis

Figure 12 and Table 3 compare the reported grid search, random search, and GA results. The GA achieved 89.30% accuracy and a reported runtime of 180.5 s, compared with 84.20% and 1250.4 s for grid search and 85.15% and 450.2 s for random search. Without verified matched search budgets and repeated runs, these results do not establish general computational or statistical superiority.

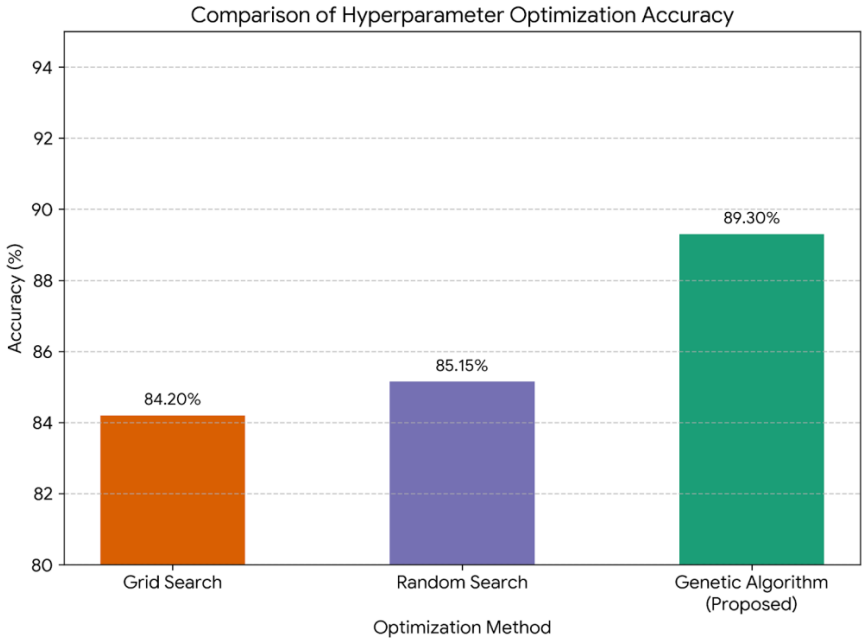


Figure 12. Comparison of hyperparameter optimization accuracy.

5. Discussion

The framework combines correlation-based predictor filtering, PSO-based majority-class instance selection, and GA-based classifier hyperparameter optimization. These operations address distinct parts of the diabetes status classification pipeline. Because the

evaluation uses cross-sectional survey data, the reported results support classification within the studied dataset, not prospective diabetes prediction or established clinical utility.

5.1. Genetic Algorithm Configuration and Reproducibility

The second stage of the proposed framework applies a Genetic Algorithm (GA) to hyperparameter optimization of the evaluated classifiers, namely the Artificial Neural Network/Multilayer Perceptron (ANN/MLP), Extreme Gradient Boosting (XGBoost), and Random Forest (RF). The GA procedure is described in terms of its evolutionary operators, including tournament selection, crossover, mutation, and termination. However, the available experimental documentation does not fully specify the population size, number of generations, crossover and mutation probabilities, chromosome encoding, model-specific hyperparameter search domains, or final selected configurations. These details are essential for reproducing the optimization procedure. In particular, continuous, integer-valued, and categorical hyperparameters require explicit encoding and decoding rules, and the selected hyperparameter configurations must be distinguished from the search ranges used during optimization. Because the original optimization logs and complete software environment records are unavailable, these settings cannot be reliably reconstructed from the reported classification metrics. Accordingly, the current study does not claim complete reproducibility of the GA optimization stage. Future experimental evaluations should document the complete GA configuration, the model-specific search spaces, the final selected hyperparameters, the software and library versions, and all random seeds used for initialization, sampling, optimization, and model training.

5.2. Limitations

The present study has several limitations. First, the framework was evaluated using a publicly available diabetes dataset, which may not fully represent diverse patient populations. Second, external validation using independent clinical datasets was not performed. Third, the proposed approach relies exclusively on structured tabular data and does not incorporate multimodal information such as medical images, continuous glucose monitoring signals, or longitudinal electronic health records. Finally, dedicated explainability analyses, including SHapley Additive exPlanations (SHAP), were not conducted. Future work should assess feature-level explanations and independent external validation. Addressing these limitations may further enhance the generalizability and clinical applicability of the proposed system.

An additional limitation of this study is the absence of a dedicated model explainability analysis. Although the proposed framework evaluates the classification performance of machine learning models, it does not include SHapley Additive exPlanations (SHAP) or other post hoc explanation methods to quantify the contributions of individual predictors to model outputs. Consequently, the present findings do not establish which survey variables contribute most strongly to individual predictions or whether the learned relationships are consistent across population subgroups. Future research could incorporate SHAP or other explainable artificial intelligence techniques to investigate feature contributions, examine model behavior, and improve the transparency of the proposed framework. Such analyses would complement, rather than replace, external validation and assessment of clinical utility.

The proposed two-stage framework involves three groups of experimental parameters: (i) parameters controlling PSO-based majority-class instance selection, (ii) parameters controlling GA-based hyperparameter optimization, and (iii) model-specific classifier configurations. For the PSO stage, the principal control parameters include swarm size, the maximum number of iterations, inertia weight, cognitive and social acceleration coefficients,

velocity bounds, the binary position conversion procedure, and the convergence criterion. These parameters govern the search for a subset of majority-class training observations. For the GA stage, the principal control parameters include population size, the maximum number of generations, tournament size, crossover and mutation probabilities, elitism strategy, and the stopping criterion. The GA also requires a defined hyperparameter search space and chromosome representation for each evaluated classifier. The final classifier configurations, dataset partitioning procedure, random seeds, and software environment are additional components required for reproducibility.

The numerical settings used in the original experiments must be obtained from the implementation records. Where these records are unavailable, the corresponding values cannot be reliably inferred from the reported classification performance. Therefore, the current study acknowledges incomplete parameter documentation as a limitation, and a fully reproducible evaluation would require verification or repetition of the experiments using explicitly recorded settings.

5.3. Clinical Implications

The proposed PSO–GA framework was evaluated using a publicly available, cross-sectional dataset derived from the Behavioral Risk Factor Surveillance System (BRFSS). The findings demonstrate the ability of the evaluated machine learning classifiers to distinguish survey-defined diabetes status within the studied dataset. However, the results should not be interpreted as evidence of prospective diabetes risk prediction, early disease detection, or direct clinical benefit.

Because the dataset does not provide longitudinal follow-up, this study cannot determine whether the framework predicts future diabetes onset. Furthermore, the analysis did not include external validation, probability calibration, decision curve analysis, or prospective clinical evaluation. Consequently, the suitability of the framework for population screening, individualized treatment planning, or clinical decision-support applications remains unestablished. Nevertheless, the proposed approach may provide a methodological basis for future research on diabetes-related classification and risk assessment. Before considering practical implementation, future studies should evaluate the framework using independent and prospective datasets, assess calibration and clinical utility, and determine whether its predictions improve decision-making or patient outcomes compared with existing approaches.

6. Conclusions

This study evaluated a two-stage PSO–GA framework for diabetes status classification using the CDC Diabetes Health Indicators dataset. PSO selected majority-class training observations to address class imbalance, and the GA optimized classifier hyperparameters. The reported results suggest improved classification performance under the evaluated conditions, particularly for minority-class discrimination. However, centroid-based instance selection can reduce sample diversity, and the results lack repeated-run uncertainty estimates, independent external validation, and prospective clinical assessment. Future studies should verify performance across random seeds, compare diversity-aware balancing methods, assess calibration, and evaluate independent and longitudinal cohorts before considering clinical deployment.

Author Contributions: Writing—original draft preparation, conceptualization, project administration, M.A.A. and F.A.; visualization, writing—review and editing, methodology, M.I.S. and F.A.; writing—review and editing, visualization and supervision, I.d.l.T.; visualization, writing—review and editing, software, M.S.G.d.M., E.S.A. and I.d.l.T. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Fundación Universidad Europea del Atlántico, Calle Isabel Torres 21, 39011 Santander, Spain (VAT: G39764972).

Data Availability Statement: The data presented in this study are openly available on https://www.kaggle.com/datasets/alexteboul/diabetes-health-indicators-dataset (accessed on 5 May 2026).

Acknowledgments: During the preparation of this manuscript, the authors used Grammarly version 14.1 and Chatgpt-6.1 for the purposes of language refinement. The authors have reviewed and edited the output and take full responsibility for the content of this publication.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

ML	Machine Learning
DL	Deep Learning
PSO	Particle Swarm Optimization
CNN	Convolutional Neural Network
SMOTE	Random Undersampling or Synthetic Oversampling
GA	Genetic Algorithm
CVD	Cardiovascular Disease
SVM	Support Vector Machine
KNN	K-Nearest Neighbor
DT	Decision Tree
CDC	Centers for Disease Control and Prevention
BRFSS	Behavioral Risk Factor Surveillance System

References

1. Alagumariappan, P.; Sathyamoorthy, M.; Dhanaraj, R.K.; Kamalanand, K.; Emmanuel, C.; Allabun, S.; Othman, M.; Getahun, M.; Soufiene, B.O. Optimized hybrid machine learning framework for early diabetes prediction using electrogastrograms. *Sci. Rep.* **2025**, *15*, 8875. [CrossRef]
2. Parkhi, D.; Periyathambi, N.; Weldeselassie, Y.G.; Patel, V.; Sukumar, N.; Siddharthan, R.; Narlikar, L.; Saravanan, P. Prediction of postpartum prediabetes by machine learning methodsS in women with gestational diabetes mellitus. *iScience* **2023**, *26*, 107846. [CrossRef]
3. Islam, M.R.; Banik, S.; Rahman, K.N.; Rahman, M.M. A comparative approach to alleviating the prevalence of diabetes mellitus using machine learning. *Comput. Methods Programs Biomed. Update* **2023**, *4*, 100113. [CrossRef]
4. Belsti, Y.; Moran, L.; Du, L.; Mousa, A.; Silva, K.D.; Enticott, J.; Teede, H. Comparison of machine learning and conventional logistic regression-based prediction models for gestational diabetes in an ethnically diverse population; the Monash GDM Machine learning model. *Int. J. Med. Inform.* **2023**, *179*, 105228. [CrossRef]
5. Khan, M.; Singh, B.K.; Nirala, N. Expert diagnostic system for detection of hypertension and diabetes mellitus using discrete wavelet decomposition of photoplethysmogram signal and machine learning technique. *Med. Nov. Technol. Devices* **2023**, *19*, 100251. [CrossRef]
6. Giorda, C.B.; Rossi, A.; Baccetti, F.; Zilich, R.; Romeo, F.; Besmir, N.; Di Cianni, G.; Guaita, G.; Morviducci, L.; Muselli, M.; et al. Achieving good metabolic control without weight gain with the systematic use of GLP-1-RAs and SGLT-2 inhibitors in type 2 diabetes: A machine learning projection using data from clinical practice. *Clin. Ther.* **2023**, *45*, 754–761. [CrossRef]
7. Allwright, M.; Karrasch, J.F.; O'Brien, J.A.; Guennewig, B.; Austin, P.J. Machine learning analysis of the UK Biobank reveals prognostic and diagnostic immune biomarkers for polyneuropathy and neuropathic pain in diabetes. *Diabetes Res. Clin. Pract.* **2023**, *201*, 110725. [CrossRef]
8. Bhatia, I.; Aarti; Ansarullah, S.I.; Amin, F.; Alabrah, A. Lightweight Advanced Deep Neural Network (DNN) Model for Early-Stage Lung Cancer Detection. *Diagnostics* **2024**, *14*, 2356. [CrossRef]
9. Riaz, S.; Lin, H.; Akhter, M.P. Design and implementation of an accelerated error convergence criterion for norm optimal iterative learning controller. *Electronics* **2020**, *9*, 1766. [CrossRef]
10. Desuky, S.; Hussain, S. An improved hybrid approach for handling class imbalance problem. *Arab. J. Sci. Eng.* **2021**, *46*, 3853–3864. [CrossRef]
11. Asnicar, F.; Thomas, A.M.; Passerini, A.; Waldron, L.; Segata, N. Machine learning for microbiologists. *Nat. Rev. Microbiol.* **2023**, *22*, 191–205. [CrossRef]
12. Wang, Y.; Song, D.; Wu, T.; Othman, E.M. Non-Fasting Glucose Measures and Their Clinical Significance in Diabetes Diagnosis and Cardiovascular and Cancer Risk Prediction: A Narrative Review. *Int. J. Mol. Sci.* **2026**, *27*, 4734. [CrossRef]

13. Christakis, C.; Saliari, D.; Zampelas, A.; Androutsos, O. Screening Tools for Early Identification of Adults at High Risk of Type 2 Diabetes: A Scoping Review. *Healthcare* **2026**, *14*, 839. [CrossRef]
14. Enríquez-Ortega, D.; Chulde-Fernández, B.; Pozo-Coral, P.; Vaca, A.; Zhinin-Vera, L.; Almeida-Galárraga, D.; Ramírez-Cando, L.; Tirado-Espín, A.; Cadena-Morejón, C.; Villalba-Meneses, F.; et al. Enhancing Diabetes Diagnosis Through Machine Learning: A Comparative Study. *Appl. Sci.* **2025**, *15*, 10087. [CrossRef]
15. Kabir, M.F.; Chen, T.; Ludwig, S.A. A performance analysis of dimensionality reduction algorithms in machine learning models for cancer prediction. *Healthc. Anal.* **2023**, *3*, 100125. [CrossRef]
16. Venkat, V.; Abdelhalim, H.; DeGroat, W.; Zeeshan, S.; Ahmed, Z. Investigating genes associated with heart failure, atrial fibrillation, and other cardiovascular diseases, and predicting disease using machine learning techniques for translational research and precision medicine. *Genomics* **2023**, *115*, 110584. [CrossRef]
17. Ahemad, M.T.; Hameed, M.A.; Vankdothu, R. COVID-19 detection and classification for machine learning methods using human genomic data. *Meas. Sens.* **2022**, *24*, 100537. [CrossRef]
18. Mujumdar, A.; Vaidehi, V. Diabetes prediction using machine learning algorithms. *Procedia Comput. Sci.* **2019**, *165*, 292–299. [CrossRef]
19. Nicolucci, A.; Romeo, L.; Bernardini, M.; Vespasiani, M.; Rossi, M.C.; Petrelli, M.; Ceriello, A.; Di Bartolo, P.; Frontoni, E.; Vespasiani, G. Prediction of complications of type 2 Diabetes: A machine learning approach. *Diabetes Res. Clin. Pract.* **2022**, *190*, 110013. [CrossRef]
20. Ganie, S.M.; Malik, M.B. An ensemble machine learning approach for predicting Type-II diabetes mellitus based on lifestyle indicators. *Healthc. Anal.* **2022**, *2*, 100092. [CrossRef]
21. Cheheltani, R.; King, N.; Lee, S.; North, B.; Kovarik, D.; Evans-Molina, C.; Leavitt, N.; Dutta, S. Predicting misdiagnosed adult-onset type 1 diabetes using machine learning. *Diabetes Res. Clin. Pract.* **2022**, *191*, 110029. [CrossRef]
22. Naz, H.; Ahuja, S. Deep learning approach for diabetes prediction using PIMA Indian dataset. *J. Diabetes Metab. Disord.* **2022**, *19*, 391–403.
23. Bhat, S.S.; Banu, M.; Ansari, G.A.; Selvam, V. A risk assessment and prediction framework for diabetes mellitus using machine learning algorithms. *Healthc. Anal.* **2023**, *4*, 100273. [CrossRef]
24. Zou, J.; Han, Y.; So, S.S. Overview of Artificial Neural Networks. In *Artificial Neural Networks: Methods in Molecular Biology™*; Livingstone, D.J., Ed.; Humana Press: Totowa, NJ, USA, 2008; Volume 458. [CrossRef]
25. Samniang, W.; Shah, S.M.T.; Tun, Y.; Munawar, A.; Sarin, K.C. A hybrid grey wolf optimized eXtreme gradient boosting-based machine learning model for hospital pharmaceutical demand forecasting. *Sci. Rep.* **2026**, *16*, 13400. [CrossRef]
26. Yang, Y.; Wang, H. Random Forest-Based Machine Failure Prediction: A Performance Comparison. *Appl. Sci.* **2025**, *15*, 8841. [CrossRef]
27. Abdulrahman, H.A.; Olawale, I.M.; Habeeb-Bello, S.; Mohammed, S.; Onumanyi, A.J.; Ajayi, O.-O. Optimizing machine learning algorithms for diabetes data: A metaheuristic approach to balancing and tuning classifiers parameters. *Frankl. Open* **2024**, *8*, 100153. [CrossRef]
28. CDC. Diabetes Health Indicators Dataset. 2015. Available online: https://www.kaggle.com/datasets/alexteboul/diabetes-health-indicators-dataset (accessed on 15 September 2026).
29. Kennedy, J.; Eberhart, R. Particle swarm optimization. In *Proceedings of ICNN'95—International Conference on Neural Networks*; IEEE: New York, NY, USA, 1995; Volume 4, pp. 1942–1948. [CrossRef]
30. Akinola, O.O.; Ezugwu, A.E.; Agushaka, J.O.; Zitar, R.A.; Abualigah, L. Multi-class feature selection with metaheuristic optimization algorithms: A review. *Neural Comput. Appl.* **2022**, *34*, 19751–19790. [CrossRef]
31. Sun, Z.; An, G.; Yang, Y.; Liu, Y. Optimized machine learning enabled intrusion detection 2 system for internet of medical things. *Frankl. Open* **2023**, *6*, 100056. [CrossRef]
32. Nawi. Prediction of hepatocellular carcinoma risk in patients with type-2 diabetes using supervised machine learning classification model. *Heliyon* **2022**, *8*, e10772. [CrossRef]
33. Tariq, U.; Ahmed, I.; Shaukat, K.; Ihsan, M. Systematic Review for Detecting & Diagnosing Brain Tumor Using Deep Learning: Current Trends and Future Research Directions. *Int. J. Comput Intell. Syst.* **2026**, *19*, 189. [CrossRef]
34. Abdullah; Siddique, A.; Shaukat, K.; Jan, T. An Intelligent Mechanism to Detect Multi-Factor Skin Cancer. *Diagnostics* **2024**, *14*, 1359. [CrossRef]
35. Ali, Z.; Hayat, M.F.; Shaukat, K.; Alam, T.M.; Hameed, I.A.; Luo, S.; Basheer, S.; Ayadi, M.; Ksibi, A. A Proposed Framework for Early Prediction of Schistosomiasis. *Diagnostics* **2022**, *12*, 3138. [CrossRef]

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.