A Hybrid QPSCO Metaheuristic for Joint Feature Selection and Hyperparameter Tuning in Early Stage Diabetes Prediction

Abdelaziz Messas*, Hichem Haouassi, Toufik Messaoud Maarouk, and Abdelouaheb Khiar

Original scientific article

***Abstract*—Type 2 Diabetes Mellitus (T2DM) continues to pose a serious health challenge. Although Machine Learning has been widely employed for disease prediction, its effectiveness depends heavily on feature selection (FS) and hyperparameter tuning (HPT). This study proposes Quantum Particle Swarm + Single Candidate Optimizer (QPSCO), a novel hybrid metaheuristic that combines the global search capability of Quantum Particle Swarm Optimization (QPSO) with the strong local exploitation ability of the Single Candidate Optimizer (SCO). The proposed approach jointly performs FS to reduce dimensionality and HPT to improve classifier performance. Experiments were conducted on the "Early Stage Diabetes Risk Prediction" dataset using several Machine Learning classifiers. The results show that QPSCO-optimized models consistently achieved competitive or superior performance in terms of accuracy, precision, F1-score, and AUC-ROC compared with related studies conducted under similar experimental conditions. QPSCO-XGBoost achieved up to 96.82% accuracy, 98.84% precision, 97.66% F1-score, and 98.32% AUC-ROC using nine selected features, while QPSCO-MLP achieved 96.42% accuracy and 98.43% AUC-ROC using only eight selected features under traditional cross-validation. Comparative evaluation showed improved performance over existing methods. Stratified Nested Cross-Validation (SNCV) with SMOTE-NC provided more reliable performance estimates. By selecting the most informative features, the proposed approach proved effective for early T2DM detection.**

***Index Terms*—Type 2 Diabetes Mellitus, Early Diagnosis, Machine Learning, Feature Selection, Hyperparameter Tuning, Metaheuristic, Quantum Particle Swarm Optimization, Single Candidate Optimizer, Stratified Nested Cross-Validation.**

I. Introduction

Diabetes is a disease characterized by abnormal regulation of glucose, protein, and lipid metabolism resulting from either an absolute or relative insulin deficiency [1]. It is typically diagnosed by elevated blood glucose levels [2]. The disease is primarily categorized into two main types: Type 1 Diabetes and Type 2 Diabetes Mellitus (T2DM). Globally, diabetes and its complications are responsible for more than 3.8 million deaths each year [3]. According to the World Health Organization (WHO), diabetes has reached epidemic proportions worldwide. In 2021, around 537 million people were living with diabetes [4]. By November 2024, this number had surpassed 800 million [5]. Early detection and timely intervention remain the most effective strategies for reducing diabetes-related complications and mortality [6].

Artificial Intelligence (AI), particularly through Machine Learning (ML) and Deep Learning (DL), has emerged as a powerful tool in computer-aided diagnosis (CAD) [7]. Numerous studies have explored AI-driven approaches for disease detection and classification, including applications in brain cancer diagnosis [8], breast cancer classification [9], coronary artery disease diagnosis [10], and early diagnosis of diabetes [11], among others. However, the predictive performance of these systems heavily depends on two interlinked factors: the relevance of selected features and the optimization of hyperparameter settings for the employed models.

With the rapid accumulation of medical data, healthcare professionals face increasing challenges in extracting relevant insights efficiently [12]. Two major tasks have thus received significant attention: Feature Selection (FS) and Hyperparameter Tuning (HPT) [13]. FS aims to identify the most informative and non-redundant features, enhancing model interpretability and reducing computational costs [14]. Meanwhile, HPT involves configuring external parameters such as learning rates, regularization strengths, or kernel functions which are not learned during model training but have a substantial impact on accuracy and generalization [15].

Both FS and HPT are NP-hard optimization problems that often involve conflicting objectives. Metaheuristics have become a practical solution for handling optimization problems [16], [17]. Most prior studies address FS and HPT as separate, sequential tasks, optimizing one while holding the other fixed. When FS and HPT are addressed independently, the interdependence between these two processes is disregarded, preventing the model from attaining its optimal configuration. It also makes the process computationally more intensive and more likely to fall into local minima.

This study is motivated by the need to enhance early and accurate T2DM diagnosis using ML models that are both efficient and easily interpretable. By jointly optimizing FS and HPT, the proposed method seeks to deliver compact yet high-performing classifiers that are better suited for real-world clinical applications.

This work proposes a novel hybrid metaheuristic method to address the joint FS–HPT problem and improve ML-based

Manuscript received February 1, 2026; revised April 13, 2026. Date of publication August 17, 2026. Date of current version August 17, 2026.

Authors are with the Department of Computer Science, ICOSI Laboratory, Abbes Laghrour University Khenchela, Algeria (e-mails: {messas.abdelaziz, haouassi.hichem, maarouk.toufik, khiar_abdelouaheb}@univ-khenchela.dz).

Digital Object Identifier (DOI): 10.24138/jcomss-2025-0293

* Corresponding author

1845-6421/09/2025-0293 © 2026 CCIS

diabetes diagnosis. The main contributions are summarized as:

- A novel hybrid metaheuristic, combining Quantum-behaved Particle Swarm Optimization with the Single Candidate Optimizer (QPSCO), is introduced to simultaneously address FS and HPT. The hybridization of QPSO's global exploration ability and SCO's efficient local refinement improves both convergence speed and robustness.
- An Optimized Latin Hypercube Sampling (OLHS) strategy is employed to generate a diverse and well-distributed initial population, ensuring a more comprehensive exploration of the search space.
- A reliable mapping method, Category-Switching Probability (CSP), translates particle positions into hyperparameter values of various types, allowing smooth transitions across categorical parameters.
- As a secondary methodological contribution, we propose an integration of SNCV with the Synthetic Minority Over-sampling Technique for Nominal and Continuous features (SMOTE-NC) algorithm oversampling confined to training folds, designed to provide a reliable evaluation under class-imbalance and feature-bias conditions

The rest of this work is organized as follows. Section II reviews the relevant literature. Section III describes the materials and methodology used in this study. In Section IV, we present the experimental setup and discuss the results through a comparative analysis. Finally, Section V concludes the study and highlights promising directions for future research.

II. Related Work

Numerous recent studies have applied ML and DL techniques to T2DM, with a particular focus on FS and HPT to improve model performance.

Several works have concentrated solely on FS techniques to identify the most relevant predictors of T2DM. For instance, Yılmaz [18] used Pearson's correlation and SelectKBest methods to find relationships between features and outcomes, realizing competitive accuracy on the 'Early Stage Diabetes Risk Prediction dataset'. Ejiyi et al. [19] applied SHAP (Shapley Additive Explanations) to determine feature importance in the PIMA dataset, reaching an accuracy of 94%. Mercaldo et al. [20] employed principal component analysis to reduce dimensionality and compared models trained on selected versus full feature sets. Ganie and Malik [21] incorporated correlation-based analysis into an ensemble framework for early stage T2DM detection.

Other research focused on optimizing model parameters without explicit FS. Wijayaningrum et al. [22] employed the improved crow search algorithm to tune MLP parameters, achieving 95.38% accuracy. Alqushaibi et al. [23] proposed a hybrid method combining Convolutional Neural Networks (CNNs) with Bayesian optimization, while [24] introduced a hybrid deep learning model combining a genetic algorithm, Stacked Autoencoder, and a Softmax classifier. Their method, which simultaneously optimized both model architecture and hyperparameters, demonstrated high predictive accuracy of 0.987 on the 'Early Stage Diabetes Risk Prediction dataset'.

A small number of studies have explored FS and HPT simultaneously. Navazi et al. [11] applied Particle Swarm Optimization (PSO) for FS and used a metaheuristic for tuning hyperparameters across Decision Tree (DT), K-Nearest Neighbors (KNN), MLP, and Support Vector Machine (SVM), with GA-optimized SVM achieving the best results (accuracy = 0.934, F1 = 0.945). Olisha et al. [25] combined Spearman correlation for FS with a grid search within a custom-designed Deep Neural Network, which significantly improved predictive accuracy on the PIMA dataset.

Other studies relied on standard classifiers or ensemble methods without addressing FS or HPT. Laila et al. [26] compared Random Forest (RF), Adaptive Boosting (AdaBoost), and Bagging, with RF achieving the best performance (accuracy = 0.971). Islam et al. [27], [28] and Nguyen et al. [29] applied common ML models such as Naïve Bayes (NB), Logistic Regression (LR), MLP, and RF across various datasets, often reporting RF as the top performer. Beyond structured datasets, [30] developed a machine-learning-based Health Cyber-Physical System (Health CPS) for early T2DM prediction using wearable IoT data, while Swapna et al. [31] applied CNN-LSTM models on ECG signals for pattern recognition. Zheng et al. [32] proposed an ML framework combining Electronic Health Records (EHR) and traditional Chinese medicine data, emphasizing multilevel feature engineering. Mansoori et al. [33] found that the Bootstrap Forest (BF) model performed best for T2DM prediction when relying on hematological factors such as the triglyceride-glucose (TyG) index.

Overall, these studies most closely align with prior work in addressing FS and HPT separately or in applying optimization techniques that are limited in their ability to balance global exploration and local refinement. To address these limitations, we propose a hybrid metaheuristic QPSCO that simultaneously performs FS and HPT by combining the global search capability of QPSO with the local exploitation strength of the Single Candidate Optimizer (SCO). The proposed approach is applied to the 'Early Stage Diabetes Risk Prediction dataset' and across nine widely used ML classifiers. Our work improves both predictive accuracy and feature reduction. Table I provides a comparative summary of related works and the distinct contributions of our study.

III. Materials and Methods

A. Dataset and Preprocessing

The dataset employed in this study is the 'Early Stage Diabetes Risk Prediction dataset', obtained from the University of California, Irvine Machine Learning Repository [34]. It was collected with direct questionnaires administered to patients at the Sylhet Diabetes Hospital in Bangladesh, and subsequently validated by qualified medical professionals. The dataset originally contains 520 instances with 16 features (age, gender, polyuria, polydipsia, sudden_weight_loss, weakness, polyphagia, genital_thrush, visual_blurring, itching, irritability, delayed_healing, partial_paresis, muscle_stiffness, alopecia, and obesity). The target variable, denoted as class, corresponds to the diagnostic outcome (Positive or Negative) and is located in the 17th column.

TABLE I
SIGNIFICANT DIFFERENCES BETWEEN THIS STUDY AND OTHERS

Papers	Duplicates removed	Feature selection	ML-HPT	Best methods	Early Stage dataset
This paper	✓	✓	✓	QPSCO-MLP	✓
Navazi et al. [11]	✓	✓	✓	GA-SVM	✓
Bulbul et al. [24]	—	F. extraction	✓	GA-SEA-Sofmax	✓
Yilmaz [18]	—	✓	—	GA-XGBoost	✓
Laila et al. [26]	—	—	—	RF	✓
Zheng et al. [32]	✓	F. summarizing	—	RF,J48,SVM	—
Mercaldo et al. [20]	—	✓	—	Hoeffding Tree	—
Ferdousi et al. [30]	—	—	—	RF	✓
Ejiyi et al. [19]	✓	F. engineering	—	XGBoost,ADAboost	—
Islam et al. [27]	—	—	—	RF	✓
Ganie et al. [21]	—	F. engineering	—	Bagged-DT	—
Wijayaningrum et al. [22]	—	—	✓	MLP	✓
Olisah et al. [25]	✓	✓	✓	GS-2GDNN	—
Nguyen et al. [29]	✓	—	✓	GS-RF	—

Upon examination, the dataset contained no missing values or evident outliers. However, 269 duplicate records were identified and removed. This is a noteworthy aspect, as duplicates could otherwise bias the learning process and artificially inflate performance metrics. After removing duplicates, 251 unique instances remained, including 173 positive cases (diabetic patients) and 78 negative cases (non-diabetic individuals), described by 16 clinical and demographic attributes, resulting in a feature-to-sample ratio of approximately 1:15.7. Furthermore, the proposed QPSCO framework incorporates feature selection before classification, reducing the final feature subset to between 5 and 10 predictors depending on the classifier. The best-performing models retained only 8–10 features, thereby substantially reducing the effective dimensionality of the learning problem and mitigating the risk of the curse of dimensionality despite the relatively limited sample size.

Given the resulting class imbalance, the SMOTE-NC algorithm [35] was applied to generate synthetic minority-class samples. Importantly, SMOTE-NC was applied only within the training folds during cross-validation, ensuring that no synthetic samples are introduced into the test folds and preventing any data leakage during evaluation.

For preprocessing, label encoding was applied to the categorical variables, and the age feature was standardized to normalize its distribution.

To explore potential interactions, we calculated a correlation matrix to assess how each pair of features relates. This approach highlights linear dependencies, helps eliminate redundant variables, and reduces the chance of overfitting [36]. As shown in Figure 1, all correlations were below 0.6, indicating that the features overlap only minimally.

During the dataset analysis, an imbalance was observed between the gender attribute and the target labels, indicating a demographic distribution bias. Additional empirical evaluations showed that retaining or removing the gender feature produced no significant impact on the performance of the proposed QPSCO-based models. Therefore, the feature was preserved in this study, while the associated dataset-level bias was explicitly acknowledged and considered within the evaluation protocol.

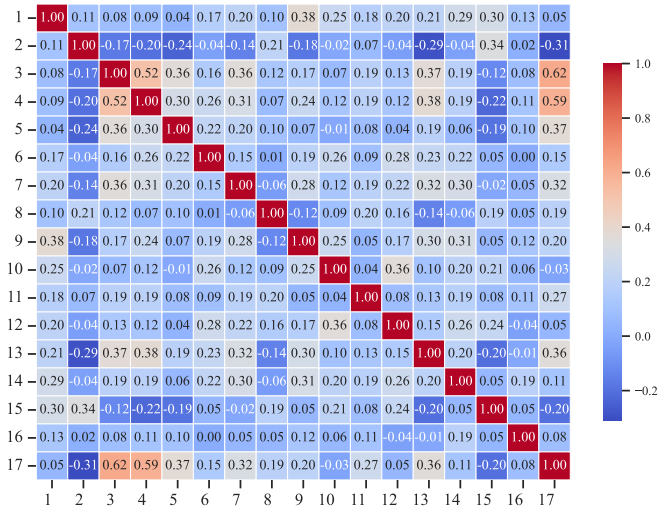


Fig. 1. Pairwise Feature Correlation Heatmap2

B. Proposed Method

In this study, we propose a hybrid optimization method that simultaneously addresses feature selection and ML hyperparameter tuning. The proposed approach combines global exploration of QPSO with local exploitation of the SCO.

To begin the search with a diverse population, we generate candidates using OLHS. QPSO drives the search to promising regions through its quantum dynamics. Upon identification of a new global best solution, SCO is activated to perform precise local refinement. QPSO and SCO keep a remarkable balance between exploitation and exploration.

The following subsections describe in detail the three main components of the proposed hybrid approach: a QPSO-based global exploration strategy; an SCO-based adaptive local refinement; and an integrated QPSCO method for joint FS and ML-HPT optimization.

1) The QPSO Algorithm : The QPSO algorithm, introduced by Sun et al. [37], is an enhanced variant of the classical PSO algorithm [38], inspired by principles of quantum mechanics. Unlike the original PSO, which updates the velocity and position of each particle, QPSO eliminates the velocity component and updates particle positions using a quantum method. This modification significantly improves the algorithm's global search ability and convergence characteristics. In QPSO, each particle i updates its position according to equations 1 and 2.

$$\boldsymbol{P}_i = \boldsymbol{\phi} \cdot \text{pbest}_i + (1 - \boldsymbol{\phi}) \cdot \text{gbest} \tag{1}$$

$$\boldsymbol{x}_i^{t+1} = \boldsymbol{P}_i + \boldsymbol{d} \cdot \beta \cdot \left|\text{mbest} - \boldsymbol{x}_i^t\right| \cdot \ln\left(\frac{1}{\boldsymbol{u}}\right) \tag{2}$$

where pbest_i is the personal best position of particle i, gbest is the global best position found by the entire swarm, $\boldsymbol{\phi} \in [0,1]^D$ is a random vector, each component independently sampled from a uniform distribution $\mathcal{U}(0,1)$, $\boldsymbol{u} \in [0,1]^D$ is another independent random vector with i.i.d. components from $\mathcal{U}(0,1)$, $\boldsymbol{d} \in \{-1,+1\}^D$ is a random direction vector, where each component is independently chosen with equal probability, β is a contraction-expansion coefficient controlling convergence behavior, $\text{mbest} = \frac{1}{N}\sum_{j=1}^{N} \text{pbest}_j$ is the mean of all particles' personal best positions, serving as a collective attractor. x_i^t is the current position of particle i at iteration t.

2) The SCO Algorithm: The SCO, introduced by Shami in 2023 [39], is a unique type of metaheuristic. Instead of relying on a large group (population) of solutions refined over time, SCO guides its search using only a single candidate solution.

The SCO algorithm manages its search by switching back and forth between exploring new areas and refining existing ones. By using a threshold α, which determines the number of iterations. SCO focuses on exploitation, making the solution updates more precise and localized. This shift is smoothed out by a monotonically decreasing weight function that gradually transitions the search behavior from exploration to exploitation. Crucially, if a candidate solution ever ventures outside the acceptable search space, it is immediately reset to the current best solution to ensure it remains valid.

The update process is mathematically formalized within the SCO method from Equation 3 to Equation 8 that define its search dynamics and solution-refinement strategy.

$$x_i = \begin{cases} gbest_i + w \cdot |gbest_i| & \text{if } r_1 < 0.5 \\ gbest_i - w \cdot |gbest_i| & \text{otherwise} \end{cases} \tag{3}$$

$$w_j = \exp\left(-\left(\frac{b \cdot j}{T_{\text{SCO}}}\right)^b\right) \tag{4}$$

$$x_i = \begin{cases} gbest_i + w \cdot r_2 \cdot (Ub_i - Lb_i) & \text{if } r_2 < 0.5 \\ gbest_i - w \cdot r_2 \cdot (Ub_i - Lb_i) & \text{otherwise} \end{cases} \tag{5}$$

$$x_i = \begin{cases} gbest_i + r_3 \cdot (Ub_i - Lb_i) & \text{if } r_3 < 0.5 \\ gbest_i - r_3 \cdot (Ub_i - Lb_i) & \text{otherwise} \end{cases} \tag{6}$$

$$x_i = \begin{cases} gbest_i & \text{if } x_i < Lb_i \text{ or } x_i > Ub_i \\ x_i & \text{otherwise} \end{cases} \tag{7}$$

$$x_i = Lb_i + r_4 \cdot (Ub_i - Lb_i) \tag{8}$$

where x_i is the i-th dimension of the candidate solution, $gbest_i$ is the i-th dimension of the global best solution, $r_1, r_2, r_3, r_4 \sim \mathcal{U}(0,1)$ are independent random numbers drawn from a uniform distribution, w is the weight controlling the step size, defined in Eq. (4), b is the user-defined shape parameter controlling the nonlinearity of the decay, j is the current SCO iteration index, and T_{SCO} is the total number of SCO iterations, Lb_i and Ub_i are the lower and upper bounds of the search space for dimension i.

3) The Hybrid QPSCO Method: To jointly address the dual problems of FS and ML-HPT, we propose a hybrid metaheuristic method that integrates the global search capability of QPSO with the adaptive local refinement of SCO.

Our search begins by using OLHS in the normalized $[0,1]^D$ space to ensure we start with a diverse, well-distributed set of initial solutions. Our QPSO strategy, which serves as the global exploration, is enhanced by its β coefficient. Crucially, whenever QPSO identifies a new global best solution ($gbest$), the SCO local refinement component is activated, executing the exploitation phase under the guidance of a low α parameter, thereby enabling precise local refinement of the current best solution without expending computational resources on extensive re-exploration.

This resulting hybridization strategy (Algorithm 1) effectively leverages the global convergence of QPSO and the improved precision in local solution refinement of SCO. This combination enables the QPSCO method to achieve a strong balance between exploration and exploitation, rendering it particularly well-adapted.

4) Encoding of Particle Positions and OLHS-Based Initialization: To initialize the optimization process, we employed OLHS [40], which produces a well-structured, stratified distribution of candidate solutions. Unlike purely random sampling, OLHS ensures a stratified distribution across each dimension of the search space. To ensure a well-distributed initial population, we employed the Maximin criterion [41]. Essentially, this technique maximizes the minimum distance between any two points. This is a strategy that promotes solution diversity and enhances overall optimization performance because it prevents initial solutions from clustering, thereby setting the stage for a more comprehensive exploration of the entire search space.

In the proposed hybrid QPSCO-based optimization method, each particle's position is encoded as a normalized continuous vector ranging from 0 to 1. The vector length equals the number of features in the dataset and the number of hyperparameters corresponding to the chosen classifier. Particle positions are updated via the QPSCO update equations.

The position vector is divided into two parts first, the FS segment composed of 16 elements, each corresponding to a feature in the dataset, where each value is passed through a transfer function to yield a binary outcome (0 for non-selected features and 1 for selected features). This mechanism allows the algorithm to construct feature subsets during the search process. Second, the HPT segment transforms each component through a dedicated mapping function into valid classifier-specific values, which may be continuous, integer, interval-bounded, discrete, or categorical.

This joint encoding mechanism allows the seamless integration of both FS and HPT within a unified QPSCO method. Figure 2 illustrates the encoding of a particle's position vector and the decoding process used to extract the FS mask and classifier-specific hyperparameters.

5) Objective Function: To evaluate a candidate solution, the position vector $X_i^t \in [0,1]^D$ is divided into two parts; the first n_f components represent the total number of the feature in the dataset, while the remaining P components encode the used classifier's hyperparameters. A general `Transfer_function()` is applied to the feature segment to generate a binary selection mask. Similarly, a `Mapping_function()` is used to interpret the hyperparameter component, translating each value into valid parameter values of different types (continuous, integer, or categorical). The resulting configuration is used to train the classifier using only the selected features.

In a first classification performance evaluation we use k-fold cross-validation and a multi-objective fitness score is defined to maximize accuracy and feature selection compactness (S_f). In a second evaluation stage, SNCV with the SMOTE-NC technique is employed alongside the traditional cross-validation (2) to provide a more robust and unbiased assessment of the models. The fitness function is calculated as in Equation (9)

$$\text{Fitness} = \omega \cdot \text{Accuracy} + (1-\omega) \cdot \left(1 - \frac{S_f}{T_f}\right) \tag{9}$$

where $\omega \in [0,1]$ is a user-defined weighting factor that controls the trade-off between classification accuracy and feature compactness, and T_f is the number of total features.

6) Transfer Functions for Binary Conversion: In the context of the FS problem, the QPSCO algorithm operates in a continuous search space, whereas the FS problem is inherently binary. To bridge this gap, a transformation mechanism known as a *transfer function* [42] is employed to convert the continuous particle position into binary values. This mechanism projects continuous search values into the range $[0,1]$ (Figure 2), after which a probabilistic thresholding scheme is applied to determine whether a feature is selected (1) or discarded (0). The method uses a fixed threshold of 0.5 presented in Equation (10), where values above the threshold are mapped to 1 and the values below to 0 [43]. However, transfer functions $T(x)$ provide more flexible and adaptive binarization, enabling smoother control over the conversion process. A widely used stochastic binarization rule is expressed as in Equation (11).

$$x_i = \begin{cases} 1 & \text{if } p_i > 0.5 \\ 0 & \text{otherwise} \end{cases} \tag{10}$$

Alternatively, transfer functions $T(x)$ allow more flexible control over the binarization process. A common stochastic binarization rule is given by:

$$x_i = \begin{cases} 1 & \text{if } T(p_i) > r, \quad r \sim \mathcal{U}(0,1) \\ 0 & \text{otherwise} \end{cases} \tag{11}$$

The most frequently used transfer functions include:

- S-shaped function (sigmoid-based).
- V-shaped function (absolute hyperbolic tangent).
- U-shaped function (symmetric parabolic form).
- M-shaped function (based on hyperbolic secant).
- Z-shaped function (inverse sigmoid).
- Fixed threshold function (hard thresholding at 0.5).

7) Mapping Function: In the context of ML HPT using the QPSCO metaheuristic, the challenge is to handle discrete and categorical hyperparameters within a continuous search space. For example, hyperparameters such as the number of trees (in RF), the maximum depth(in XGBoost), or the activation function (in MLP) cannot be directly modeled as continuous variables.

To overcome this incompatibility, discrete hyperparameters are managed through rounding operations (e.g., `round()` or `int()`) and by clipping (`clip()`) to keep the values in valid bounds. For categorical hyperparameters, the mapping function assigns each category a unique index from a predefined set. This index-based encoding enables categorical parameters to be integrated into the continuous search process of population-based metaheuristics (Figure 2).

However, categorical variables introduce additional complexity due to their unordered or partially ordered discrete nature, which hinders smooth and continuous exploration of the solution space. We propose a probabilistic mapping based on a category transition mechanism, controlled by a novel concept referred to as the Category-Switching Probability (CSP), denoted as P_{switch}. The CSP mechanism determines whether a categorical value should remain unchanged or be switched to another category during a particle update.

Unlike traditional deterministic mappings, the CSP framework enables a more controlled and adaptive exploration of the categorical search space.

Formally, at each iteration t, for any categorical hyperparameter $x_c \in \mathcal{C}$, where $\mathcal{C}$ is the finite set of possible categories, the update rule is defined as:

$$x_c^{(t+1)} = \begin{cases} x_c^{(t)} & \text{prob. } 1 - P_{\text{switch}}, \\ \text{Expl}(x_c^{(t)}) & \text{prob. } P_{\text{switch}}, \end{cases} \tag{12}$$

where the function `Expl` selects a new category according to one of the following strategies:

- Uniform random exploration: a new category $x_c' \in \mathcal{C} \setminus \{x_c^{(t)}\}$ is selected uniformly at random, enabling a complete transition away from the current value.

- Neighborhood-based exploration (for ordered categories): a neighboring category is chosen such that $d(x_c, x'_c) \leq 1$ according to a predefined discrete distance metric, allowing smooth and localized transitions.

Algorithm 1 QPSCO-based Optimization for FS and HPT

Input: Dataset (X, y), classifier type, number of features D, number of particles N, maximum iterations T, SCO parameters (b, T_{sco})
Output: Best solution (gbest), selected features, best model, and performance metrics
Initialize particle positions $\boldsymbol{x}_i$, $i = 1, \ldots, N$ with OLHS
Initialize personal bests: $\text{pbest}_i \leftarrow \boldsymbol{x}_i$, scores $f(\text{pbest}_i)$
  Initialize global best gbest $\leftarrow \emptyset$ (or null)
  for each iteration $t = 1$ to T **do**
    for each particle $i = 1$ to N **do**
      Evaluate fitness $f(\boldsymbol{x}_i)$ using Algorithm 2
      Update personal best:
      if $f(\boldsymbol{x}_i) > f(\text{pbest}_i)$ **then**
        $\text{pbest}_i \leftarrow \boldsymbol{x}_i$
      end if
      Update global best:
      if $f(\boldsymbol{x}_i) > f(\text{gbest})$ **then**
        gbest $\leftarrow \boldsymbol{x}_i$
        Store best model, selected features, metrics
        Initialize SCO parameters (α, counters)
        for $j = 1$ to T_{sco} **do**
          Compute adaptive weight w
          Generate $\boldsymbol{x}$ using SCO update rules
          Apply boundary control to $\boldsymbol{x}$
          Evaluate fitness $f(\boldsymbol{x})$
          if $f(\boldsymbol{x}) > f(\text{pbest}_i)$ **then**
            Update $\text{pbest}_i \leftarrow \boldsymbol{x}$
          end if
          if $f(\boldsymbol{x}) > f(\text{gbest})$ **then**
            Update gbest $\leftarrow \boldsymbol{x}$
            Update best model and metrics
          end if
        end for
      end if
    end for
    Update particle positions using QPSO update rule
  end for
  Return gbest, best model, selected features, metrics

The CSP solution introduces and guarantees a real transition into the bounds values of categorical parameters. To further promote convergence, the switching probability P_{switch} is progressively reduced over time according to a decay schedule that gradually shifts the search behavior from exploration to exploitation. A commonly adopted decay strategy can be expressed by Equations (13) and (14).

$$P_{\text{switch}}(t) = P_0 \cdot \exp(-\lambda \cdot t) \tag{13}$$

or, alternatively, a linear decay strategy:

$$P_{\text{switch}}(t) = P_0 \cdot \left(1 - \frac{t}{T_{\text{max}}}\right), \quad \text{for } t \leq T_{\text{max}}, \tag{14}$$

where P_0 is the initial switching probability, t is the current iteration, λ is the decay rate, and T_{max} is the maximum number of iterations.

Algorithm 2 Fitness Function with Dynamic SMOTE-NC and SNCV Evaluation

Input: Particle $\boldsymbol{x}$, dataset (X, y), number of features D
Output: score $f(\boldsymbol{x})$, Selected features and metrics
Split X_i into feature segment X_f and H-Paramater segment X_h
Apply `Transfer_function`(X_f)
Apply `Mapping_function`(X_h)
Subset the dataset using FS mask
Construct stratification labels
Ensure resampling is applied only on training folds to prevent data leakage
Perform stratified K-fold cross-validation on X_s
for each fold $(\mathcal{D}_{train}, \mathcal{D}_{val})$ **do**
  Split data: $(X_{tr}, y_{tr}), (X_{val}, y_{val})$
  Apply SMOTE/SMOTENC only on training set
  Train model $\mathcal{M}$ on $(X_{tr}^{res}, y_{tr}^{res})$
  Predict labels $\hat{y}$ on validation set
  Compute probability scores $\hat{p}$ if available
end for
Aggregate metrics over all folds
Compute fitness using Eq. (9)
Return $f(\boldsymbol{x})$, selected features, and evaluation metrics

8) Evaluation Protocol: In this study, a clear distinction is made between benchmarking performance using traditional cross-validation [44] and deployment-oriented performance estimation using SNCV. Medical datasets are often affected by class imbalance and feature-level biases, which may compromise the reliability of model evaluation if not properly addressed. In particular, the dataset used in this work exhibits both an imbalanced target distribution and a non-uniform gender representation.

To mitigate these issues, SMOTE-NC was applied exclusively within the training folds, thereby preventing any data leakage that could result from performing oversampling prior to data partitioning, in line with recent best practices. The nested design further ensures a strict separation between hyperparameter optimization and final performance estimation, thus eliminating selection bias. Within this framework, the outer loop is used for unbiased evaluation, while the inner loop is dedicated to QPSCO-based optimization restricted to the training data only [45].

Stratification is enforced across both loops to preserve the original class and gender distributions within each fold [46]. This design leads to a bias-aware and statistically reliable evaluation protocol that more accurately reflects the generalization capability of the proposed models on unseen clinical data.

IV. Experimental Studies and Results

A. Experimental Studies

1) Configuration Details: The proposed QPSCO algorithm demonstrated rapid convergence (Figure 3) when applied to optimize relatively simple classifiers such as SVM, NB, DT, LR, ADAboost and KNN. Satisfactory performance was typically achieved within 50 iterations using a swarm size of 30 particles for QPSO and 20 iterations for SCO (Table II).

To further evaluate the adequacy of the available sample size, a post-hoc learning curve analysis was performed using progressively larger stratified subsets containing 50, 100, 150, 200, and 251 samples. The complete SNCV-QPSCO optimization process was repeated for each subset, and both the fitness score and F1-score were recorded.

As shown in Fig. 4, both metrics improve steadily as the sample size increases and reach a stable region around 200 samples. Beyond this point, only marginal performance variations are observed, while the standard deviations continue to decrease. This behavior suggests that the proposed framework has reached a performance plateau under the current feature space and that the available dataset size is sufficient to obtain stable and reproducible predictive performance within the adopted SNCV protocol. However, more complex classifiers like RF, XGBoost and MLP required substantially more iterations (up to 90) to reach convergence. This difference in convergence speed reflects the impact of the classifier's internal complexity and the dimensionality of their hyperparameter search space. Accordingly, a differentiated optimisation strategy was adopted, where the maximum number of iterations was dynamically adjusted based on the classifier's complexity. This ensured optimal allocation of optimization effort for each model.

After several empirical tests, the weight parameter ω of the objective function was empirically fixed at 0.95, providing a balance between model accuracy and the number of selected features. The coefficient β in QPSO was set to 0.9, to encourage exploration during the search, the SCO algorithm was configured with a minimum α value to enhance exploitation. Furthermore, the parameter b in SCO was set to 1.5, as this value yielded superior performance during preliminary experimental evaluations.

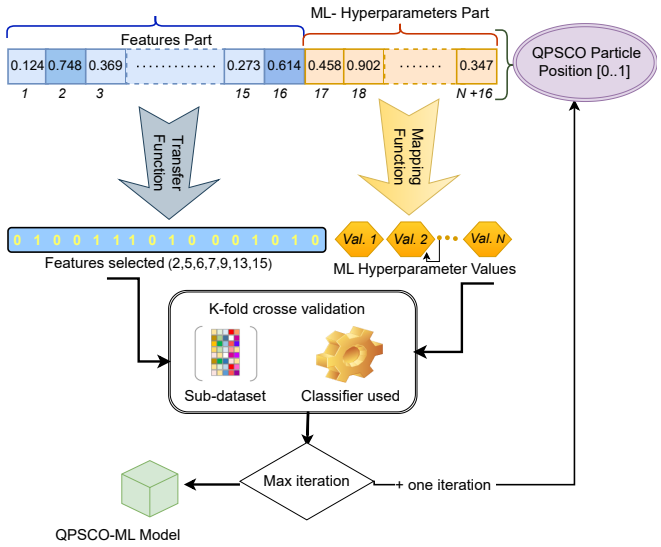


Fig. 2. QPSCO Agent Position Processing

In the context of binary feature selection, several transfer functions were also examined, including the classical S-shaped, V-shaped, U-shaped, Mshaped, Z-shaped, and Gaussian forms. Notably, a fixed-threshold binarization function using a fixed threshold of 0.5, consistently produced superior results in terms of both accuracy and convergence stability compared to the other transfer functions.

To ensure a rigorous and bias-aware evaluation, the SNCV scheme was adopted, consisting of 5 outer folds for unbiased performance estimation and 3 inner folds dedicated to hyperparameter optimization via the proposed metaheuristic. Stratification was enforced to preserve the original class and gender distribution in each fold. To address the dataset's class imbalance (173 positive vs. 78 negative instances), during each outer SNCV iteration, SMOTE-NC was applied exclusively to the training folds. The augmentation ratio was conservatively limited to a maximum of 25% of the cleaned training partition (approximately 201 samples), thereby reducing the class imbalance while avoiding excessive oversampling.

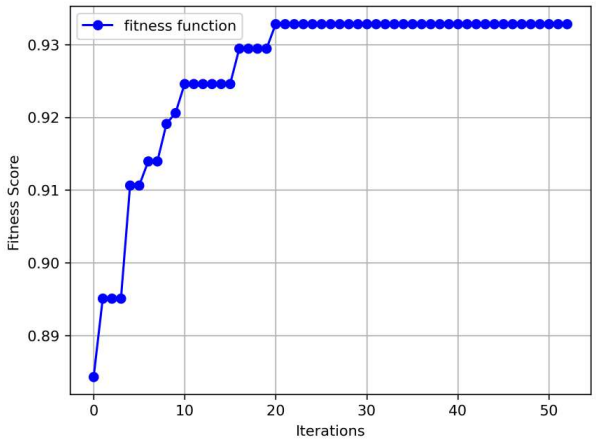


Fig. 3. QPSCO-SVM Convergence

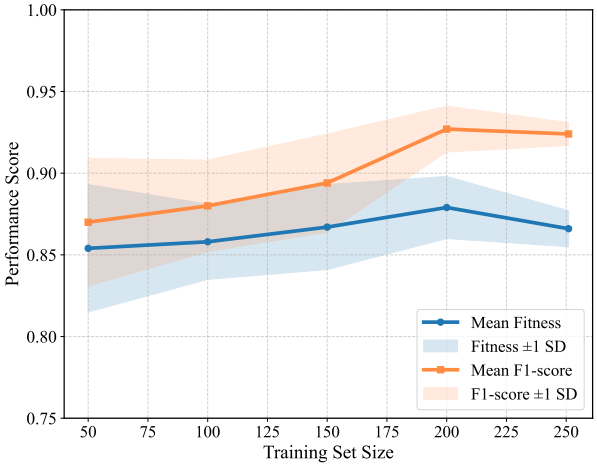


Fig. 4. Learning Curve of the SVM Classifier under the Proposed SNCV-SMOTE Protocol

TABLE II
QPSCO PARAMETER SETTING

Parameter	Values	Selected
QPSO particles	10, 20, 30, 40, 50	30
QPSO iterations	30, 40, 50, 80, 100	50
SCO iterations	10, 15, 20, 30	20
β	0.5, 0.9, 1.0	0.9
b	0.8, 1.5, 3.0	1.5
ω	0.6, 0.8, 0.9, 0.95	0.95

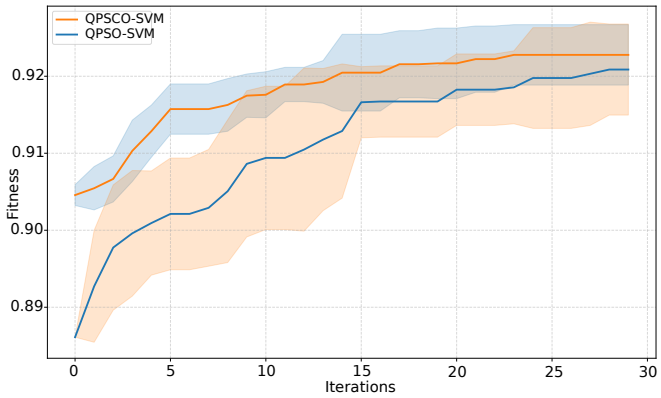


Fig. 5. Convergence curves of hybrid QPSCO and standard QPSO

B. Results

1) Effectiveness of the QPSCO Hybrid Model and OLHS Initialization Strategy: To validate the effectiveness of the proposed QPSCO hybrid metaheuristic, we first compared its performance against the standard QPSO algorithm, which does not incorporate the SCO component. Both methods were evaluated over several independent optimization runs conducted under identical experimental settings.

As illustrated in Figure 5, the average convergence curve of QPSCO consistently remains above that of QPSO, demonstrating faster convergence and higher final fitness values. This performance can be attributed to the integration of the SCO algorithm, which enhances the swarm's exploitation. The results demonstrate that QPSCO effectively mitigates premature convergence, enabling the swarm to escape local optima and better explore the search space.

Beyond the hybridization effect, we also examined the influence of the initialization strategy on optimization performance. Specifically, the OLHS initialization method was compared to conventional random initialization. Two-dimensional visualizations of the initial positions of 30 particles generated by each approach reveal a distinct difference in distribution. Random initialization (Figure 7) produces a scattered yet clustered configuration, whereas OLHS (Figure 6) achieves a uniform and well-stratified coverage of the search domain.

As illustrated in Figure 8, the convergence curve for OLHS-based initialization consistently surpasses that of random initialization across multiple runs. The enhanced diversity and spatial coverage offered by OLHS allow the swarm to explore more regions of the search space and improve the reliability of the search. These findings confirm that combining the QPSCO hybrid model with OLHS initialization leads to a more efficient, stable, and high-quality optimization process, providing strong evidence of the method's overall effectiveness.

2) Optimal Hyperparameters and Features Selected: Table III summarizes the optimal hyperparameter configurations obtained for each of the nine evaluated machine learning classifiers after 30 independent optimization runs. For each model, the corresponding subset of selected features is also reported. These results reflect the configurations that yielded performance.

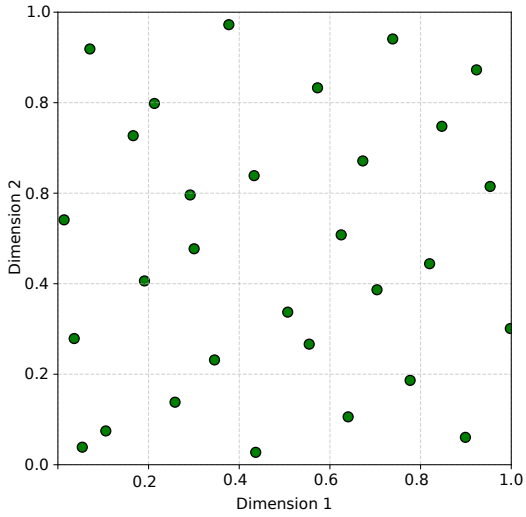


Fig. 6. Well-distributed initial population OLHS with 30 particles

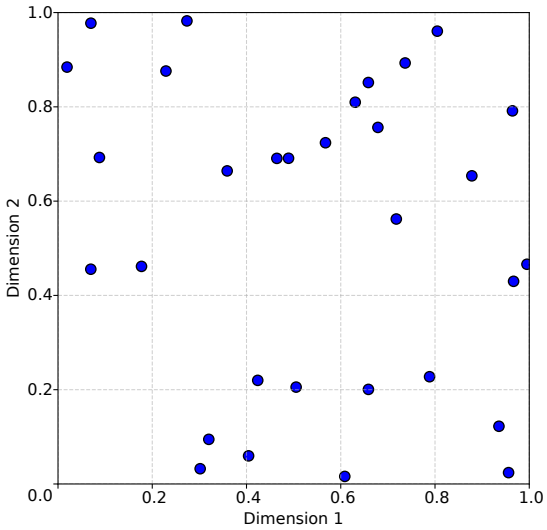


Fig. 7. Distributed initial population Random with 30 particles

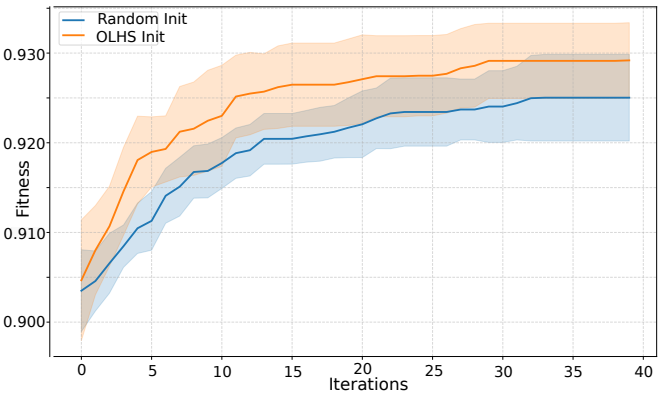


Fig. 8. Convergence SVM with OLHS and Random Initialization

3) Classifiers Performance with QPSCO Optimization Using Cross-Validation: To comprehensively evaluate the proposed QPSCO-based hybrid metaheuristic, the experimental results are organized into two complementary phases. In the

first phase, the proposed method was applied with standard cross-validation and without any resampling, across two versions of the dataset: the original preprocessed dataset including duplicates (520 instances) and the cleaned dataset with duplicates removed (251 instances). This phase aimed to benchmark the proposed approach against well-established studies in the Type 2 Diabetes prediction literature using the same nine ML classifiers. In the second phase, the evaluation was conducted exclusively on the cleaned dataset (251 instances) using the proposed SNCV evaluation framework (5 outer / 3 inner folds) with SMOTE-NC applied dynamically within the training folds, assessing the robustness of the proposed metaheuristic under controlled conditions that account for class imbalance and feature bias. To ensure statistical reliability and mitigate the stochastic nature of the optimization process, all reported results represent the average performance over 30 independent runs for each of the nine classifiers, allowing for a stable and reproducible estimation of the proposed method's generalization capability. Across both phases, results are reported in terms of the most discriminative performance metrics — Accuracy, Recall, F1-score, AUC-ROC, Specificity — alongside the feature reduction rate achieved by the proposed selection mechanism. Table **??** presents the performance results of the proposed method, averaged over 30 runs. All classifiers improved both accuracy and feature reduction, with the MLP classifier as the top-performing model, achieving the highest average accuracy of 95.35% ± 0.87 and a best accuracy of 96.42%, along with strong F1 and specificity scores. The RF, SVM, and XGBoost classifiers follow closely, achieving accuracies of 95.45%, 95.44%, and 94.28%, respectively, with best accuracies of 95.62% for SVM and 96.82% for XGBoost. These results exhibit low variability across runs, indicating stable and reproducible behavior of the proposed learning framework under the SNCV evaluation protocol.

Beyond the top-performing models, other classifiers demonstrated interesting trade-offs between feature reduction and predictive performance. NB and LR, in particular, achieved the lowest average number of selected features, with means of 5.53 and 5.70, respectively (Figure 9), while still attaining respectable average accuracies of 89.29% and 90.01%.

4) Impact of Class Imbalance Handling Strategies: To evaluate the impact of class imbalance handling, experiments were conducted comparing SMOTE-NC, class weighting, and no balancing strategies (Table IV). SMOTE-NC achieved the best overall performance in terms of fitness (87.77%), accuracy (90.79%), and F1-score (93.29%), while class weighting produced comparable results and the highest ROC-AUC (95.17%). The differences among the three strategies are marginal.

These findings suggest that the performance of the proposed framework is primarily driven by the joint FS and HPT process, while remaining stable across different reasonable imbalance-handling techniques, highlighting its robustness within the evaluated experimental setting.

5) Statistical Significance and Ranking Analysis of Classifiers: To rigorously evaluate the comparative performance of the nine classifiers, we conducted a series of non-parametric statistical tests. The Friedman test [47] was first applied to assess whether significant differences existed among the nine evaluated classifiers' performance distributions. The resulting p-value, well below the 0.05 threshold, confirmed the presence of significant overall differences.

Subsequently, pairwise significance testing was conducted using the Mann–Whitney U test [48] combined with the Holm correction for multiple comparisons [49], as illustrated in Figure 10. The heatmap of corrected p-values, where red cells indicate statistically differences and blue cells denote non-significant differences. Significance levels are represented as follows $p < 0.001$ (highly significant), $p < 0.01$ (moderately significant), and $p < 0.05$ (significant).

The analysis reveals that the SVM outperforms several other classifiers, including RF, DT, KNN, LR, NB, and XGBoost, with statistically significant differences ($p < 0.001$ in most comparisons). Furthermore, RF also performs better than DT, KNN, MLP, LR, and NB, with strong statistical evidence. Comparisons between XGBoost and AdaBoost, as

TABLE III
OPTIMAL HYPERPARAMETER RANGES AND FEATURES SELECTED

ML	Hyper-parameter	Optimal Value	FeaturesSelected
DT	max_depth	27	1,2,3,4,8,11,12,15
	min_samples_split	4	
	min_samples_leaf	1	
	criterion	entropy	
RF	n_estimators	456	1,2,3,4,8,9,10,12,15,16
	max_depth	32	
	min_samples_leaf	1	
	min_samples_split	2	
SVM	C	8.995567	1,2,3,4,9,12,14,15
	kernel	rbf	
	gamma	0.923121	
KNN	n_neighbors	8	2,3,4,8,9,11,14,16
	metric P	euclidean	
	leaf_size	40	
LR	C	5.7375	2,3,4,13,15
	penalty	l2	
	solver	lbfgs	
	max_iter	252	
NB	var_smoothing	0.000154	2,3,4,5,6
ADA	n_estimators	226	2,3,4,5,10,15
	learning_rate	0.633694824	
XGB	n_estimators	203	1,2,3,4,8,9,10,14
	max_depth	3	
	learning_rate	0.158574114	
	gamma	0.111300166	
	subsample	0.687959577	
	colsample_bytree	0.935924143	
	min_child_weight	1	
	reg_alpha	0.183904017	
MLP	num_hidden_layers	2	1,2,3,4,8,9,10,14,16
	hidden_layer1_size	27	
	hidden_layer2_size	50	
	activation	relu	
	alpha	0.002490153	
	max_iter	300	
	solver	lbfgs	

well as between DT and XGB, show no statistically significant differences ($p > 0.05$). The analysis indicates that MLP, XGBoost, RF and SVM are the best-performing classifiers for maximizing predictive accuracy, while LR and NB are less suitable when predictive precision is a primary objective.

TABLE IV
PERFORMANCE COMPARISON OF CLASS IMBALANCE HANDLING STRATEGIES (QPSCO-SVM)

Strategy	*Fitness (%)*	*Accuracy (%)*	*F1-S (%)*	*AUC (%)*
No Balancing	87.60±1.14	90.32±1.07	92.99±0.81	94.68±1.80
Class Weighting	87.71±1.32	90.69±1.18	93.21±0.89	95.17±1.68
SMOTE-NC	87.77±1.44	90.79±1.37	93.29±1.02	95.01±1.76

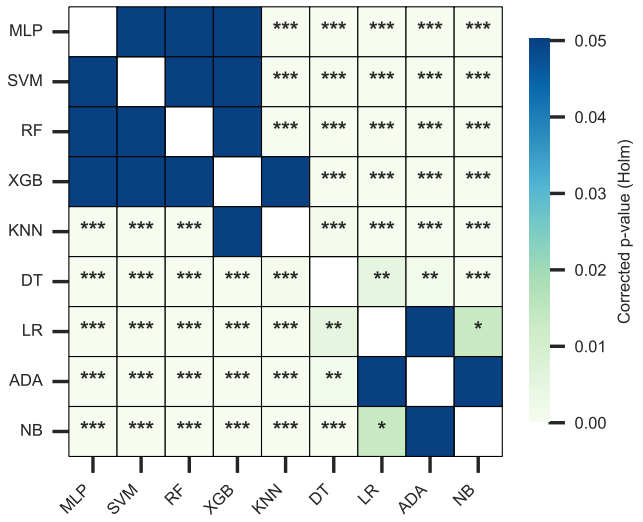


Fig. 10. Pairwise Statistical Significance (Mann–Whitney U, Holm correction)

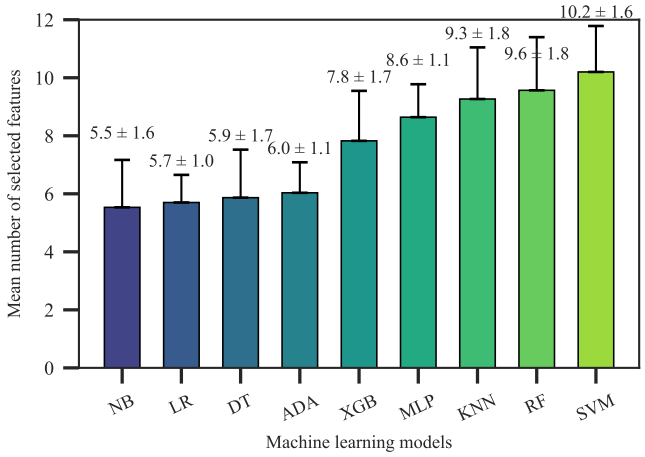


Fig. 9. Number of Selected Features by QPSCO-ML Models

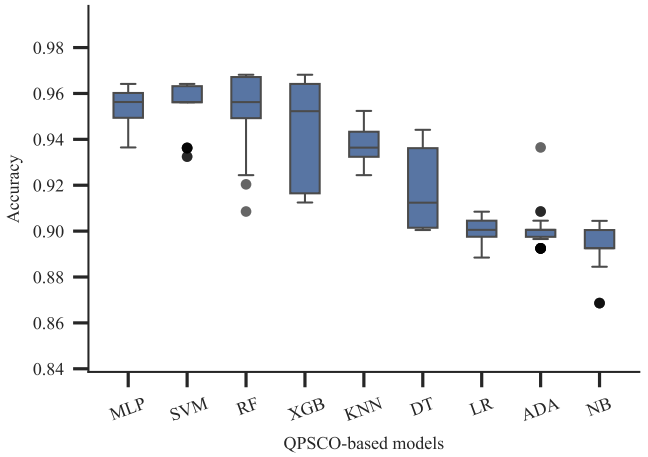


Fig. 11. Boxplots of QPSCO-Based Model Performance

6) Performance Stability and Variability Across Models: The boxplot analysis [50] (Figure 11) conducted across 30 independent runs highlights notable differences in the stability. The XGB model exhibits the largest interquartile range (IQR), indicating higher variability and greater sensitivity to stochastic factors during optimization. In contrast, both MLP and SVM exhibit narrower IQRs, indicating high stability and performance, with SVM showing no outliers.

While MLP demonstrates generally reliable behavior, XGBoost and LR distributions are right-skewed, with upper whiskers extending beyond the boxes, suggesting that some runs achieved exceptionally high results relative to the median.

Overall, this analysis shows that SVM and MLP deliver the most stable yet high-performing results, whereas XGB and LR exhibit greater variability, and AdaBoost and Naive Bayes demonstrate the highest reliability at the expense of lower predictive accuracy.

The graph in Figure 12 reveals a reduction in fitness performance when using SNCV with SMOTE-NC compared to traditional 5-fold Cross-Validation. This reduction is expected, as SNCV applies a stricter and more reliable evaluation protocol that prevents information leakage between training and testing phases, leading to more realistic estimates of generalization performance.

In contrast, traditional cross-validation often yields overly optimistic results due to implicit overfitting during feature selection and hyperparameter optimization. The use of SMOTE-NC also increases the learning complexity by generating synthetic minority samples, forcing the models to learn more generalizable decision boundaries rather than overfitting to dominant class patterns.

Although an overall decrease in performance is observed under SNCV, the relative ranking of classifiers remained largely consistent, with RF, SVM, and MLP still achieving the best performance. This confirms the stability and reliability of the proposed QPSCO-based optimization framework under rigorous evaluation conditions.

TABLE V
HYBRID QPSCO-ML CLASSIFIER RESULTS USING CROSS-VALIDATION

QPSCO-ML Classifier	Fitness*(%)	Accuracy(%)	Precision(%)	Recall(%)	F1 Score(%)	Specificity(%)	AUC-ROC(%)	Selected Features
MLP Mean	92.88 ± 0.77	95.35 ± 0.87	97.27 ± 0.73	96.02 ± 1.33	96.60 ± 0.67	93.93 ± 1.66	97.41 ± 0.84	8.64 ± 1.14
— Best	94.09	96.42	97.70	97.13	97.40	94.83	98.43	8
RF Mean	92.69 ± 1.12	95.45 ± 1.52	97.32 ± 1.55	96.11 ± 1.09	96.68 ± 1.10	94.08 ± 3.65	97.33 ± 1.29	9.57 ± 1.83
— Best	93.34	95.62	97.65	95.98	96.79	94.92	96.33	8
SVM Mean	92.48 ± 0.90	95.44 ± 1.04	97.30 ± 1.22	96.12 ± 0.86	96.67 ± 0.75	93.97 ± 2.84	97.38 ± 1.31	10.20 ± 1.58
— Best	93.03	95.62	97.65	95.95	96.79	94.92	97.90	9
XGB Mean	92.12 ± 1.71	94.28 ± 2.21	96.11 ± 2.61	95.76 ± 1.20	95.86 ± 1.56	91.14 ± 6.08	96.76 ± 1.64	7.83 ± 1.72
— Best	94.17	96.82	98.84	96.55	97.66	97.50	98.32	9
KNN Mean	91.23 ± 0.52	93.82 ± 0.84	94.53 ± 1.20	96.75 ± 1.20	95.57 ± 0.59	87.45 ± 3.02	94.20 ± 2.03	9.27 ± 1.78
— Best	91.83	94.03	93.47	98.24	95.77	84.83	91.53	8
DT Mean	90.45 ± 1.26	91.87 ± 1.75	95.66 ± 2.84	92.66 ± 1.50	94.01 ± 1.19	90.33 ± 6.66	93.14 ± 1.44	5.87 ± 1.66
— Best	92.51	94.42	98.78	93.08	95.78	97.50	94.72	7
LR Mean	88.73 ± 0.47	90.01 ± 0.61	91.81 ± 0.70	94.24 ± 0.63	92.88 ± 0.39	81.00 ± 2.10	94.40 ± 0.57	5.70 ± 0.95
— Best	89.74	90.85	91.91	95.41	93.51	81.08	94.10	5
ADA Mean	88.55 ± 0.27	89.93 ± 0.46	91.80 ± 0.75	94.12 ± 0.61	92.81 ± 0.26	80.99 ± 2.24	94.77 ± 0.62	6.03 ± 1.05
— Best	89.12	90.85	92.32	94.84	93.46	82.33	95.48	7
NB Mean	88.10 ± 0.76	89.29 ± 0.98	92.96 ± 1.35	91.64 ± 2.57	92.14 ± 0.87	84.34 ± 3.65	93.27 ± 1.31	5.53 ± 1.63
— Best	89.67	90.45	92.30	94.25	93.15	82.33	94.14	4

* $Fitness = \omega \cdot Accuracy + (1 - \omega) \cdot \left(1 - \frac{S_f}{16}\right)$

TABLE VI
HYBRID QPSCO-ML CLASSIFIER RESULTS USING SNCV AND SMOTE-NC

QPSCO-ML Classifier	Fitness*(%)	Accuracy(%)	Precision(%)	Recall(%)	F1 Score(%)	Specificity(%)	AUC-ROC(%)	Selected Features
RF Mean	87.92 ± 1.92	90.85 ± 1.76	94.97 ± 1.00	91.70 ± 1.85	93.22 ± 1.37	89.14 ± 2.08	95.28 ± 1.71	10.86 ± 1.07
— Best	89.68	92.43	95.97	93.09	94.41	91.17	95.37	10
SVM Mean	87.60 ± 1.14	90.32 ± 1.07	92.95 ± 1.09	93.23 ± 1.64	92.99 ± 0.81	83.98 ± 2.77	94.68 ± 1.80	10.27 ± 1.72
— Best	89.21	91.27	94.14	93.11	93.55	87.33	94.21	8
MLP Mean	87.11 ± 1.59	89.26 ± 1.71	94.02 ± 1.22	90.33 ± 1.99	92.02 ± 1.33	87.04 ± 2.75	94.40 ± 1.62	8.60 ± 1.19
— Best	90.02	92.45	95.95	93.14	94.39	91.17	97.78	9
XGB Mean	86.50 ± 1.45	88.62 ± 1.47	94.06 ± 1.08	89.33 ± 1.71	91.45 ± 1.15	87.26 ± 2.43	93.96 ± 1.19	8.63 ± 1.39
— Best	88.80	90.84	94.81	91.97	93.16	88.42	94.46	8
ADA Mean	85.20 ± 1.14	86.45 ± 1.04	91.73 ± 1.26	89.00 ± 0.67	89.99 ± 0.69	81.29 ± 3.48	92.70 ± 0.99	6.17 ± 1.47
— Best	86.39	87.65	92.57	89.66	90.86	83.58	93.93	6
NB Mean	84.61 ± 1.18	85.74 ± 1.20	91.96 ± 1.15	87.10 ± 1.26	89.33 ± 0.89	82.85 ± 2.70	91.30 ± 0.86	5.90 ± 1.21
— Best	86.72	87.66	92.78	89.09	90.80	84.67	92.61	5
KNN Mean	84.57 ± 1.61	86.65 ± 1.71	90.71 ± 1.42	90.28 ± 2.14	90.25 ± 1.30	78.89 ± 3.57	90.27 ± 2.19	8.77 ± 1.63
— Best	88.04	90.04	92.32	93.66	92.83	82.33	94.09	8
DT Mean	84.52 ± 1.99	86.20 ± 2.08	92.92 ± 1.88	86.97 ± 2.15	89.59 ± 1.57	84.71 ± 4.40	90.97 ± 2.11	7.60 ± 1.04
— Best	88.02	90.02	95.19	90.17	92.50	89.67	92.73	8
LR Mean	83.51 ± 1.81	84.56 ± 1.70	91.72 ± 1.24	85.60 ± 1.91	88.38 ± 1.34	82.51 ± 2.86	92.19 ± 1.34	5.83 ± 1.46
— Best	87.08	88.05	93.38	89.03	91.12	86.00	94.47	5

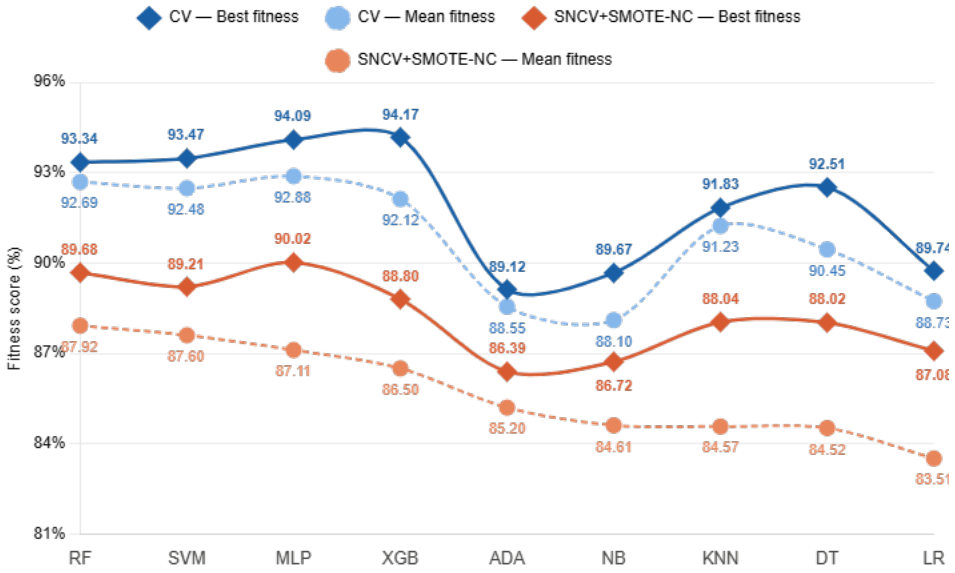


Fig. 12. Comparative Fitness Evaluation of QPSCO-Based Classifiers Using 5-Fold Cross-Validation and SNCV with SMOTE-NC

TABLE VII
EXPERIMENTAL COMPARISON WITH RELATED WORK ON THE SAME DATASET WITHOUT DUPLICATES

Study	Method	Accuracy(%)	Precision(%)	Recall(%)	F1-Score(%)	Specificity(%)	AUC-ROC(%)	Selected Features
Navazi et al. [11]	GA-SVM	93.42	95.56	93.48	94.51	93.33	93.41	9
Our model.	QPSCO-XGB	96.82	98.84	96.55	97.66	97.50	98.32	9
Our model.	QPSCO-MLP	96.42	97.70	97.13	97.40	94.83	98.43	8
Our model.	QPSCO-RF	95.62	97.65	95.98	96.79	94.92	96.33	8
Our model.	QPSCO-SVM	95.62	97.65	95.95	96.79	94.92	97.90	9

TABLE VIII
EXPERIMENTAL COMPARISON WITH RELATED WORK ON THE SAME DATASET WITHE DUPLICATES

Study	Method	Accuracy(%)	Precision(%)	Recall(%)	F1 Score(%)	Specificity(%)	AUC-ROC(%)	Selected Features
Ferdousi et al. [30]	RF	94.00	100	85.18	92.00	100	—	16
Wijayaningrum et al. [22]	MLP	95.38	—	—	—	—	—	16
Laila at al. [26]	RF	97.11	97.10	97.10	97.10	—	—	16
Yılmaz [18]	GA-XGBoost	97.23	—	—	97.00	—	—	10
islam et al. [27]	RF	97.40	—	—	—	—	—	16
Bulbul et al. [24]	GA-SAE-Softmax	98.72	98.11	98.11	98.11	99.02	—	16
Our Model	QPSCO-RF	98.46	98.45	99.06	98.75	97.50	98.54	7
Our Model	QPSCO-XGB	98.46	99.06	98.44	98.75	98.50	99.11	8
Our Model	QPSCO-MLP	98.08	97.87	99.06	98.45	96.50	99.28	9
Our Model	QPSCO-SVM	98.08	96.98	100.00	98.46	95.00	98.30	9

C. Comparative Evaluation and Discussion

As shown in Table VII, our proposed QPSCO-MLP, QPSCO-XGBoost, QPSCO-RF and QPSCO-SVM models outperform the GA-SVM method by Navazi et al. (2023) [11] under identical preprocessing and evaluation conditions when available, across all major evaluation metrics, including accuracy (96.42%, 96.82% and 95.62% vs. 93.42%), recall, F1 score, and AUC-ROC. Importantly, both studies employed the 'Early Stage Diabetes Risk Prediction dataset' (without duplicate samples). Furthermore, some of our models achieve these superior results while using fewer features (8 vs. 9), demonstrating higher efficiency and a more compact feature representation.

In Table VIII, the same set of models are compared with

several recent studies that, like ours, used the"Early Stage Diabetes Risk Prediction" dataset, which contains duplicated records. Our method achieved 98.46% accuracy with QPSCO-RF and QPSCO-XGBoost and 98.08% with QPSCO-MLP and QPSCO-SVM, all using only 7 to 9 selected features. These results outperform most existing methods, such as GA-XGBoost (97.23%) and Random Forest (97.40%). The only study reporting slightly higher accuracy, Bulbul et al. [24] (98.72%), employed a much more complex deep hybrid architecture (GA-SAE-Softmax) with all 16 features. This comparison further underscores the efficiency and predictive strength of our proposed QPSCO-based method.

Although the use of SNCV evaluation with SMOTE-NC reduced the overall performance compared with traditional cross-validation, the observed performance reduction is attributable to two complementary factors: the nested cross-validation structure eliminates optimistic bias inherent in standard evaluation, while the application of SMOTE-NC increases the complexity of the learned decision boundaries by exposing the models to synthetically balanced class distributions. Consequently, the resulting estimates, although more conservative, are statistically more reliable and better reflect true generalization capability. Unlike conventional validation methods, SNCV maintains a clear separation between hyperparameter tuning and final testing while preserving class distribution in each fold. As a result, it helps reduce overfitting and provides more reliable performance estimates, especially for imbalanced medical datasets. Therefore, despite the slight reduction in results, this evaluation strategy improves the fairness and credibility of the experimental findings.

V. Conclusion

This study proposes a novel hybrid metaheuristic, QPSCO, designed for the simultaneous optimization of FS and HPT for early detection of T2DM. By combining the global search capability of QPSO with the strong local refinement mechanism of SCO, the proposed method achieves an effective balance between global exploration and local exploitation within the solution search space. The experimental results clearly demonstrate that the ML models optimized using QPSCO consistently surpass conventional models. Moreover, when compared with established models such as MLP ,XG-Boost, SVM and Random Forest, QPSCO achieved results that were comparable or superior while relying on a significantly reduced subset of features. These results clearly show that the proposed approach is solid and reliable, while also helping to achieve better accuracy and a clearer understanding of how the model works, both of which are critical factors in medical diagnosis. To further strengthen the reliability of the evaluation, a SNCV evaluation was adopted, combined with SMOTE-NC applied exclusively within training folds. This protocol provides a more rigorous assessment of the proposed models and confirms the robustness of QPSCO under realistic and clinically relevant conditions. Future work will focus on applying QPSCO to larger and more diverse datasets to improve its generalization in real clinical settings and make it a promising tool for computer-aided diagnosis. Additionally, the extension of QPSCO to other domains, including computer vision, natural language processing, and cybersecurity will be investigated.

References

[1] A. C. Nnamudi, N. J. Orhue, I. I. Ijeh, and A. N. Nwabueze, "Finnish diabetes risk score outperformed triglyceride-glucose index in diabetes risk prediction," *Journal of Diabetes & Metabolic Disorders*, vol. 22, no. 2, pp. 1337–1345, 2023, doi: 10.1007/s40200-023-01252-y.

[2] H. Cheng, J. Zhu, P. Li, and H. Xu, "Combining knowledge extension with convolution neural network for diabetes prediction," *Engineering Applications of Artificial Intelligence*, vol. 125, p. 106658, 2023, doi: 10.1016/j.engappai.2023.106658.

[3] N. Barakat, A. P. Bradley, and M. N. H. Barakat, "Intelligible support vector machines for diagnosis of diabetes mellitus," *IEEE Transactions on Information Technology in Biomedicine*, vol. 14, no. 4, pp. 1114–1120, 2010, doi: 10.1109/TITB.2009.2039485.

[4] A. Kumar, R. Gangwar, A. A. Zargar, R. Kumar, and A. Sharma, "Prevalence of diabetes in india: A review of idf diabetes atlas 10th edition," *Current Diabetes Reviews*, vol. 20, no. 1, pp. 105–114, 2024, doi: 10.2174/1573399819666230413094200.

[5] World Health Organization, "Urgent action needed as global diabetes cases increase four-fold over past decades," 2024, news release. [Online]. Available: https://www.who.int/news/item/13-11-2024-urgent-action-needed-as-global-diabetes-cases-increase-four-fold-over-past-decades

[6] V. V. Vijayan and C. Anjali, "Prediction and diagnosis of diabetes mellitus—a machine learning approach," in *2015 IEEE Recent Advances in Intelligent Computational Systems (RAICS)*, 2015, pp. 122–127, doi: 10.1109/RAICS.2015.7488400.

[7] G. Chugh, S. Kumar, and N. Singh, "Survey on machine learning and deep learning applications in breast cancer diagnosis," *Cognitive Computation*, vol. 13, no. 6, pp. 1451–1470, 2021, doi: 10.1007/s12559-020-09813-6.

[8] M. Aljohani, W. M. Bahgat, H. M. Balaha, Y. AbdulAzeem, M. El-Abd, M. Badawy, and M. A. Elhosseini, "An automated metaheuristic-optimized approach for diagnosing and classifying brain tumors based on a convolutional neural network," *Results in Engineering*, vol. 23, p. 102459, 2024, doi: 10.1016/j.rineng.2024.102459.

[9] S. Thirumalaisamy, K. Thangavilou, H. Rajadurai, O. Saidani, N. Alturki, S. K. Mathivanan, and S. Gochhait, "Breast cancer classification using synthesized deep learning model with metaheuristic optimization algorithm," *Diagnostics*, vol. 13, no. 18, p. 2925, 2023, doi: 10.3390/diagnostics13182925.

[10] H. Haouassi, R. Mahdaoui, O. Chouhal, and A. Bekhouche, "An efficient classification rule generation for coronary artery disease diagnosis using a novel discrete equilibrium optimizer algorithm," *Journal of Intelligent & Fuzzy Systems*, vol. 43, no. 3, pp. 2315–2331, 2022, doi: 10.3233/JIFS-213257.

[11] F. Navazi, Y. Yuan, and N. Archer, "An examination of the hybrid meta-heuristic machine learning algorithms for early diagnosis of type ii diabetes using big data feature selection," *Healthcare Analytics*, vol. 4, p. 100227, 2023, doi: 10.1016/j.health.2023.100227.

[12] W. Li, G.-G. Wang, and A. H. Gandomi, "A survey of learning-based intelligent optimization algorithms," *Archives of Computational Methods in Engineering*, pp. 1–19, 2021, doi: 10.1007/s11831-021-09579-z.

[13] R. Abu Khurmaa, I. Aljarah, and A. Sharieh, "An intelligent feature selection approach based on moth flame optimization for medical diagnosis," *Neural Computing and Applications*, vol. 33, pp. 7165–7204, 2021, doi: 10.1007/s00521-020-05483-5.

[14] V. V. Kolisetty and D. S. Rajput, "A review on the significance of machine learning for data analysis in big data," *Jordanian Journal of Computers and Information Technology*, vol. 6, no. 1, 2020, doi: 10.5455/jjcit.71-1564729835.

[15] Y. Li, Y. Zhang, and Y. Cai, "A new hyperparameter optimization method for power load forecast based on recurrent neural networks," *Algorithms*, vol. 14, no. 6, p. 163, 2021, doi: 10.3390/a14060163.

[16] S. Kaur, Y. Kumar, A. Koul, and S. Kumar Kamboj, "A systematic review on metaheuristic optimization techniques for feature selections in disease diagnosis: Open issues and challenges," *Archives of Computational Methods in Engineering*, vol. 30, no. 3, pp. 1863–1895, 2023, doi: 10.1007/s11831-022-09853-1.

[17] J. Singh, J. K. Sandhu, and Y. Kumar, "Metaheuristic-based hyperparameter optimization for multi-disease detection and diagnosis in machine learning," *Service Oriented Computing and Applications*, vol. 18, no. 2, pp. 163–182, 2024, doi: 10.1007/s11761-023-00382-8.

[18] A. Yılmaz, "Prediction of type 2 diabetes mellitus using feature selection-based machine learning algorithms," *Health Problems of Civilization*, vol. 16, no. 2, pp. 128–139, 2022, doi: 10.5114/hpc.2022.114541.

[19] C. J. Ejiyi, Z. Qin, J. Amos, M. B. Ejiyi, A. Nnani, T. U. Ejiyi, V. K. Agbesi, C. Diokpo, and C. Okpara, "A robust predictive diagnosis model for diabetes mellitus using shapley-incorporated machine learning algorithms," *Healthcare Analytics*, vol. 3, p. 100166, 2023, doi: 10.1016/j.health.2023.100166.

[20] F. Mercaldo, V. Nardone, and A. Santone, "Diabetes mellitus affected patients classification and diagnosis through machine learning techniques," *Procedia Computer Science*, vol. 112, pp. 2519–2528, 2017, doi: 10.1016/j.procs.2017.08.193.

[21] S. M. Ganie and M. B. Malik, "An ensemble machine learning approach for predicting type-ii diabetes mellitus based on lifestyle indicators," *Healthcare Analytics*, vol. 2, p. 100092, 2022, doi: 10.1016/j.health.2022.100092.

[22] V. Wijayaningrum, T. Saragih, and N. Putriwijaya, "Optimal multilayer perceptron parameters for early stage diabetes risk prediction," in *IOP Conference Series: Materials Science and Engineering*, vol. 1073, no. 1. IOP Publishing, 2021, p. 012070, doi: 10.1088/1757-899X/1073/1/012070.

[23] A. Alqushaibi, M. H. Hasan, S. J. Abdulkadir, A. Muneer, M. Gamal, Q. Al-Tashi, S. M. Taib, and H. Alhussian, "Type 2 diabetes risk prediction using deep convolutional neural network based-bayesian optimization," *Computational Materials and Continua*, vol. 75, no. 2, 2023, doi: 10.32604/cmc.2023.035655.

[24] M. A. BULBUL, "A novel hybrid deep learning model for early stage diabetes risk prediction," *The Journal of Supercomputing*, vol. 80, no. 13, pp. 19 462–19 484, 2024, doi: 10.1007/s11227-024-06211-9.

[25] C. C. Olisah, L. Smith, and M. Smith, "Diabetes mellitus prediction and diagnosis from a data preprocessing and machine learning perspective," *Computer Methods and Programs in Biomedicine*, vol. 220, p. 106773, 2022, doi: 10.1016/j.cmpb.2022.106773.

[26] U. E. Laila, K. Mahboob, A. W. Khan, F. Khan, and W. Taekeun, "An ensemble approach to predict early-stage diabetes risk using machine learning: An empirical study," *Sensors*, vol. 22, no. 15, p. 5247, 2022, doi: 10.3390/s22145247.

[27] M. F. Islam, R. Ferdousi, S. Rahman, and H. Y. Bushra, "Likelihood prediction of diabetes at early stage using data mining techniques," in *Computer Vision and Machine Intelligence in Medical Image Analysis: International Symposium, ISCMM 2019*. Springer, 2020, pp. 113–125, doi: 10.1007/978-981-13-8798-2_12.

[28] M. Islam, J. Rahman, M. Abedin, M. Ali, and N. A. M. F. Ahmed, "Identification of the risk factors of type 2 diabetes and its prediction using machine learning techniques," *Health Systems*, vol. 12, pp. 243–254, 2023, doi: 10.1016/j.ijcce.2022.02.002.

[29] L. P. Nguyen, D. D. Tung, D. T. Nguyen, H. N. Le, T. Q. Tran, T. V. Binh, and D. T. N. Pham, "The utilization of machine learning algorithms for assisting physicians in the diagnosis of diabetes," *Diagnostics*, vol. 13, no. 12, p. 2087, 2023, doi: 10.3390/diagnostics13122087.

[30] R. Ferdousi, M. A. Hossain, and A. El Saddik, "Early-stage risk prediction of non-communicable disease using machine learning in health cps," *IEEE Access*, vol. 9, pp. 96 823–96 837, 2021, doi: 10.1109/ACCESS.2021.3094063.

[31] G. Swapna, S. Kp, and R. Vinayakumar, "Automated detection of diabetes using cnn and cnn-lstm network and heart rate signals," *Procedia Computer Science*, vol. 132, pp. 1253–1262, 2018, doi: 10.1016/j.procs.2018.05.041.

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

[33] A. Mansoori, T. Sahranavard, Z. S. Hosseini, S. S. Soflaei, N. Emrani, E. Nazar, M. Gharizadeh, Z. Khorasanchi, S. Effati, M. Ghamsary, G. Ferns, H. Esmaily, and M. G. Mobarhan, "Prediction of type 2 diabetes mellitus using hematological factors based on machine learning approaches: A cohort study analysis," *Scientific Reports*, vol. 13, pp. 1–11, 2023, doi: 10.1038/s41598-022-27340-2.

[34] "Early stage diabetes risk prediction [dataset]," UCI Machine Learning Repository, 2020, doi: 10.24432/C5VG8H. [Online]. Available: https://archive.ics.uci.edu/dataset/529/early+stage+diabetes+risk+prediction+dataset

[35] E. C. Gök and M. O. Olgun, "Smote-nc and gradient boosting imputation based random forest classifier for predicting severity level of covid-19 patients with blood samples," *Neural Computing and Applications*, vol. 33, no. 22, pp. 15 693–15 707, 2021, doi: 10.1007/s00521-021-06189-y.

[36] M. Kuhn, K. Johnson *et al.*, *Applied predictive modeling*. Springer, 2013, vol. 26, doi: 10.1007/978-1-4614-6849-3.

[37] J. Sun, W. Xu, and B. Feng, "A global search strategy of quantum-behaved particle swarm optimization," in *Proceedings of the 2004 IEEE Conference on Cybernetics and Intelligent Systems*. IEEE, 2004, pp. 111–116, doi: 10.1109/ICCIS.2004.1460396.

[38] J. Sun, B. Feng, and W. Xu, "Particle swarm optimization with particles having quantum behavior," in *Proceedings of the 2004 congress on evolutionary computation (IEEE Cat. No. 04TH8753)*, vol. 1. IEEE, 2004, pp. 325–331, doi: 10.1109/CEC.2004.1330875.

[39] T. M. Shami, "Single candidate optimizer: A novel optimization algorithm," *Soft Computing*, 2023, doi: 10.1007/s12065-022-00762-7.

[40] M. D. McKay, R. J. Beckman, and W. J. Conover, "A comparison of three methods for selecting values of input variables in the analysis of output from a computer code," *Technometrics*, vol. 42, no. 1, pp. 55–61, 2000, doi: 10.1080/00401706.1979.10489755.

[41] M. D. Morris and T. J. Mitchell, "Exploratory designs for computational experiments," *Journal of statistical planning and inference*, vol. 43, no. 3, pp. 381–402, 1995, doi: 10.1016/0378-3758(94)00035-T.

[42] A. Khiar, S. Mazouzi, R. Benaboud, and H. Haouassi, "Binary tuna swarm optimization algorithm-based feature selection for intrusion detection systems," *Journal of Communications Software and Systems*, vol. 21, no. 4, pp. 383–393, 2025, doi: 10.24138/jcomss-2025-0083.

[43] H.-M. Song, Y.-C. Wang, J.-S. Wang, Y.-W. Song, S. Li, Y.-L. Qi, and J.-N. Hou, "Dynamic time-varying transfer function for cancer gene expression data feature selection problem," *Journal of Big Data*, vol. 12, no. 1, p. 53, 2025, doi: 10.1186/s40537-025-01105-w.

[44] S. Bates, T. Hastie, and R. Tibshirani, "Cross-validation: what does it estimate and how well does it do it?" *Journal of the American Statistical Association*, vol. 119, no. 546, pp. 1434–1445, 2024, doi: 10.1080/01621459.2023.2197686.

[45] M. J. Lewis, A. Spiliopoulou, K. Goldmann, C. Pitzalis, P. McKeigue, and M. R. Barnes, "nestedcv: an r package for fast implementation of nested cross-validation with embedded feature selection designed for transcriptomics and high-dimensional data," *Bioinformatics advances*, vol. 3, no. 1, p. vbad048, 2023, doi: 10.1093/bioadv/vbad048.

[46] S. Szeghalmy and A. Fazekas, "A comparative study of the use of stratified cross-validation and distribution-balanced stratified cross-validation in imbalanced learning," *Sensors*, vol. 23, no. 4, p. 2333, 2023, doi: 10.3390/s23042333.

[47] M. Friedman, "The use of ranks to avoid the assumption of normality implicit in the analysis of variance," *Journal of the American Statistical Association*, vol. 32, no. 200, pp. 675–701, 1937, doi: 10.1080/01621459.1937.10503522.

[48] H. B. Mann and D. R. Whitney, "On a test of whether one of two random variables is stochastically larger than the other," *The annals of mathematical statistics*, pp. 50–60, 1947, doi: 10.1214/aoms/1177730491.

[49] S. Holm, "A simple sequentially rejective multiple test procedure," *Scandinavian Journal of Statistics*, vol. 6, no. 2, pp. 65–70, 1979. [Online]. Available: https://www.jstor.org/stable/4615733

[50] R. McGill, J. W. Tukey, and W. A. Larsen, "Variations of box plots," *The american statistician*, vol. 32, no. 1, pp. 12–16, 1978, doi: 10.1080/00031305.1978.10479236.



Abdelaziz Messas is a State Engineer in Computer Science graduated from the University of Annaba in 1996. He obtained his Master's degree in Computer Science from the University of Khenchela in 2012. Since 2024, he has been pursuing a Ph.D. in Artificial Intelligence at the University of Khenchela. His research interests focus on the development of intelligent systems, machine learning, methaheuristics and swarm-based optimization, and the application of AI techniques in medical and scientific domains.



Hichem Haouassi received his Ph.D. degree in computer science from the University of Batna, Algeria in 2012, and now he is a full professor in Abbas Laghrour University, Khenchela, Algeria. His main research interests are the Artificial intelligence, Data mining, Methaheuristics and swarm-based optimization, Feature selection, and classification.



Toufik Messaoud Maarouk is a Professor in the Department of Computer Science, Faculty of Sciences and Technology, at the University of Khenchela, Algeria. He received his Ph.D. in Computer Science from Constantine University, Algeria, in 2012. He currently serves as the Director of the ICOSI Laboratory . His research focuses on formal methods for the specification and verification of concurrent and distributed systems.



Abdelouaheb Khiar received his Ph.D. degree in computer science from the University of Oum El Bouaghi,in 2026, He conducted research at the ICOSI Laboratory, Abbes Laghrour University, Khenchela. Since 2012, he has served as an Assistant Professor at Abbes Laghrour University, where he contributes to research in artificial intelligence and cybersecurity. His work focuses on metaheuristic optimization, cybersecurity, and machine learning applications for network security.