



OPEN

A hybrid framework for heart disease prediction using classical and quantum-inspired machine learning techniques

Ankur Kumar¹, Sanjay Dhanka², Abhinav Sharma², Rohit Bansal^{3,4}, Mochammad Fahlevi^{5,6}✉, Fazla Rabby⁷ & Mohammed Aljuaid⁸✉

This research proposes a novel framework for enhancing heart disease prediction using a hybrid approach that integrates classical and quantum-inspired machine learning techniques. The framework leverages a combined dataset comprising Cleveland, Hungarian, Switzerland, Long Beach, and Statlog datasets, encompassing 1190 observations. After preprocessing and removing 272 duplicate entries, the final dataset consists of 918 unique observations. Data preprocessing has been performed to handle missing values, outliers, and correlations. Feature selection has been employed to identify the most relevant attributes for heart disease prediction. Subsequently, both classical and quantum-inspired models are trained and optimized. The classical models utilized Genetic Algorithms (CGA) and Particle Swarm Optimization (CPSO) for hyperparameter tuning, while the quantum-inspired models employed Quantum Genetic Algorithms (QGAs) and Quantum Particle Swarm Optimization (QPSO). A Support Vector Machine (SVM) classifier has been used in both classical and quantum domains. Tenfold cross-validation has been performed to assess model performance using metrics such as accuracy, F1-score, precision, sensitivity, specificity, positive likelihood ratio (LR+), negative likelihood ratio (LR-), and diagnostic odds ratio (DOR). The performance of the classical and quantum models has been compared to existing state-of-the-art approaches. The results demonstrated the potential of the proposed hybrid framework in achieving improved heart disease prediction accuracy and robustness.

Keywords Machine learning, Arrhythmia classification, Particle swarm optimization, Feature selection, Pearson's correlation coefficient

The heart plays a crucial role in the human body, as life hinges on its efficient functioning. When the heart does not operate properly, it can negatively impact other vital organs, including the brain and kidneys. Acting as a pump, the heart circulates blood throughout the body. Insufficient blood flow can cause various organs, particularly the brain, to suffer. If the heart ceases to beat, death can occur within minutes. The overall effectiveness of the heart is essential for sustaining life. Heart disease encompasses a range of conditions and issues that can affect the heart's performance. The World Health Organization (WHO) projects that by 2030, heart disease will cause nearly 23.6 million deaths annually. Many fatalities occur because heart disease often goes undetected in its early stages. Early prediction of the disease can significantly reduce the number of deaths. This prediction relies on recognizing symptoms, conducting physical examinations, and identifying clinical signs of heart disease, which include various functional and pathological factors. Numerous researchers have identified several factors that elevate the risk of heart disease. Major risk factors include smoking, tobacco use, high blood cholesterol, diabetes mellitus, obesity, and hypertension, all of which can heighten the likelihood of developing coronary heart and

¹School of Computing and Electrical Engineering (SCEE), Indian Institute of Technology (IIT) Mandi, Mandi 175005, Himachal Pradesh, India. ²Department of Electrical and Instrumentation Engineering, Sant Longowal Institute of Engineering and Technology, Longowal, Sangrur, Punjab, India. ³Department of Management Studies, Vaish College of Engineering, Rohtak, India. ⁴Adjunct Faculty, Rockford College, Sydney, Australia. ⁵Management Department, BINUS Online, Bina Nusantara University, Jakarta 11480, Indonesia. ⁶Operation Research & Management Sciences, Faculty of Business and Management, Universiti Sultan Zainal Abidin (UniSZA), Kuala Terengganu, Malaysia. ⁷Stanford Institute of Management and Technology, Parramatta, Australia. ⁸Department of Health Administration, College of Business Administration, King Saud University, Riyadh, Saudi Arabia. ✉email: mochammad.fahlevi@binus.ac.id; maljuaid@ksu.edu.sa

blood vessel diseases. Additionally, contributing risk factors such as stress and alcohol consumption may also increase the risk of heart disease, although their exact significance and prevalence are not yet fully understood. Many of these risk factors can be modified, controlled, or managed. Consequently, individuals with more risk factors have a higher chance of experiencing coronary artery heart disease (CAHD).

In modern days, the introduction of machines and artificial intelligence (AI) into the medical field has led to notable advancements and improvements. One significant issue in cardiology is assessing a person's risk of heart failure. Utilizing multiple longitudinal study auto-regression analyses has helped in creating predictive methods for this purpose. However, the advancements in technology have also led to the generation of enormous amounts of data in healthcare databases, making it difficult to analyze and interpret the information accurately. Machine learning and quantum machine learning study how computers learn from observation and experience. Machine learning algorithms have the potential to address various issues in medical centre management and data analysis, facilitating the interpretation of large datasets. Heart disease prediction involves several factors, including age, cholesterol levels, weight, height, sex, blood pressure, resting ECG (electrocardiogram), chest pain, smoking, obesity, and dietary habits. A major challenge in this field is the extensive number of features used in predicting heart disease, which complicates the task. The numerous features also make classification in machine learning more difficult, affecting performance, and reducing the accuracy of machine learning systems. Therefore, overcoming these challenges is essential for enhancing heart disease diagnosis.

The proposed hybrid framework leverages feature selection and interpretability to bridge the gap between machine learning and clinical practice. By combining classical (CGA, CPSO) and quantum-inspired (QGA, QPSO) techniques, the model identifies the most predictive features while maintaining transparency for clinicians. This is critical in healthcare, where clinicians require clear, actionable insights to trust AI-driven predictions. The reduced feature set (19% with CGA, 37% with QGA) not only improves computational efficiency but also highlights clinically relevant biomarkers—such as chest pain type, resting BP, and cholesterol levels—that align with established diagnostic criteria. This approach ensures the model is both accurate and interpretable, facilitating its adoption in real-time diagnostic decision-making and personalized treatment planning.

Literature survey

In the realm of medical science, classification stands out as one of the most vital, significant, and widely utilized decision-making tools. In the pursuit of accurately and effectively predicting coronary heart disease, numerous modern technologies have been introduced. Here, we provide a brief overview of some of the relevant works in this area. Numerous soft computing methods have been developed for diagnosing heart disease, which generally consists of two primary components: feature selection and classification. The feature selection process involves identifying a smaller set of relevant features that effectively represent the dataset while removing redundant and irrelevant attributes. Using CGA and QGA, the model prioritizes clinically interpretable biomarkers (e.g., chest pain type, resting BP, ST depression) that align with established diagnostic protocols (e.g., ACC/AHA guidelines). This ensures the selected features are not only statistically significant but also actionable for clinicians. These chosen features are then used in the classification component, where patients are categorized into specific classes based on the selected features. The quantum-inspired QGA further optimizes this process by leveraging superposition to evaluate feature combinations more efficiently than classical methods, reducing the final feature set by 37% without compromising accuracy. Mohen et al.¹ proposed a novel approach was developed to enhance the accuracy of predicting heart disease by detecting significant features using machine-learning techniques. This strategy involved employing various combinations of features and multiple classification techniques for prediction. The researchers achieved an accuracy of 88.7% using the hybrid random forest with a linear model (HRFLM). Long et al.² combined rough sets-based attribute reduction using the chaos firefly algorithm with an interval type-2 fuzzy logic system for heart disease diagnosis, outperforming other classifiers in accuracy and efficiency. While effective, the approach struggles with large datasets and slow training times. Future research will focus on heuristic methods, enhancing the chaos firefly algorithm with Levy flights, and employing parallel algorithms to improve speed. Dwivedi et al.³ applied six machine learning models—Artificial Neural Network (ANN), SVM, Logistic Regression (LR), K-Nearest Neighbor (KNN), classification tree, and NB—were compared using various performance metrics. The results indicated that LR outperformed the other models, achieving accuracy, sensitivity, specificity, and precision scores of 85%, 89%, 81%, and 85%, respectively. Reddy et al.⁴ applied Several machine learning classifiers—including Decision Tree (DT), Discriminant Analysis, LR, Naïve Bayes (NB), SVM, KNN, Bagged Trees, Optimizable Tree, and Optimizable KNNs—were trained using 10-fold cross-validation. The models were tested on the Correlation-based Feature Selection (CFS) optimal set of the dataset. The Optimizable KNNs algorithm achieved a top accuracy of 95.04% and an AUC of 0.99 with CFS, compared to 90.34% accuracy and 0.96 AUC with full features. Heidari et al.⁵ used a dataset of 1190 observations from various sources and trained multiple machine learning classifiers. Using 10-fold cross-validation on a CFS optimal set, the Optimizable Hybrid Quantum Neural Random Forest (HQRF) algorithm achieved 90.52% and 94.31% accuracies on Statlog and Cleveland datasets, respectively. Table 1 illustrates additional state-of-the-art advancements in the field.

The effectiveness of data mining techniques in predicting cardiovascular disease relies heavily on the selection of key features and the appropriate use of machine learning algorithms. Identifying the optimal combination of significant features and the best-performing algorithm is crucial. This research aims to determine the most effective data mining techniques and feature sets for heart disease prediction. However, current methods are often insufficient, highlighting the need for thorough evaluation and experimentation to identify the best techniques and features for accurate predictions.

Studies	Preprocessing/Feature Selection Methods	Classifiers	Accuracy (%)	Sensitivity (%)	Limitation
Ezekiel et al. ⁶	Chi-square	SVM	85	88	Validation of external datasets required
Paul et al. ⁷	Data normalization	ANN	88.47	---	Only accuracy has been evaluated other performance metrics could be explored
Ayon et al. ⁸	Correlation	LR, DT, NB, RF	92.41, 92.76, 91.18, 89.41	94, 96, 90, 87	High Model complexity and less interpretability
Gullu et al. ⁹	GA and Tabu search	SVM	80.5	80.5	Validation of external datasets required
Sarra et al. ¹⁰	Chi-square	SVM	89.7	89.74	The findings should be validated
Heidari et al. ⁵	Data normalization	Hybrid Quantum Neural Random Forest (HQRF)	90.52	---	Ensemble methods for hybridization have not been discussed
Nijaguna et al. ¹¹	Quantum Fruit Fly Algorithm (QFFA)	SVM	99.04	99.26	ResNet50 and VGG16 could be computationally complex.
Amraoui et al. ¹²	---	Quantum Random Forest (QRF)	91	---	Further validation on larger and more diverse datasets is needed.
Sridevi et al. ¹³	---	Quantum Transfer Learning (QTL)	84	---	validation efforts should be focused
Shi et al. ¹⁴	Pearson correlations	Quantum Random Forest (QRF)	82.1	81.8	A relatively small sample size of 50 IPF subjects.
Wang et al. ¹⁵	Quantum principal component analysis (QPCA)	Quantum deep neural networks (QDNN)	90.87	---	Further validation on larger and more diverse datasets is needed.
Dutta et al. ¹⁶	Principal component analysis (PCA)	DT	81.86	89	There is a need for data pre-processing steps.
Sreejith et al. ¹⁷	Red Deer Algorithm (RDA)	RF	89.81	89.73	The assumptions made during modeling should be mentioned.
Nasim et al. ¹⁸	Chi-squared based	Gradient Boosting Classier (GBC)	89.5	89.5	Overfitting and validation should be discussed

Table 1. State of the art.

S. No.	Feature	Description
1.	Age	Age of the patient [years]
2.	Sex	Sex of the patient [M: Male, F: Female]
3.	Chest Pain Type	Chest pain type [TA: Typical Angina, ATA: Atypical Angina, NAP: Non-Anginal Pain, ASY: Asymptomatic]
4.	Resting BP	Resting blood pressure [mm Hg]
5.	Cholesterol	Serum cholesterol [mm/dl]
6.	Fasting BS	Fasting blood sugar [1: if FastingBS > 120 mg/dl, 0: otherwise]
7.	Resting ECG	Resting electrocardiogram results [Normal: Normal, ST: having ST-T wave abnormality, LVH: probable or definite LVH]
8.	Max HR	Maximum heart rate achieved [Numeric value between 60 and 202]
9.	Exercise Angina	Exercise-induced angina [Y: Yes, N: No]
10.	Oldpeak	ST [Numeric value measured in depression]
11.	ST_Slope	The slope of the peak exercise ST segment [Up: upsloping, Flat: flat, Down: downsloping]
12.	Heart Disease	Output class [1: heart disease, 0: Normal]

Table 2. Data attributes description.

Materials and methods

Data description

The heart disease datasets used in this research are publicly available from UCI repository. This comprehensive heart disease dataset was created by merging five previously independent datasets, making it the largest of its kind available for research purposes. The combined dataset includes 11 common features as listed in Table 2 and encompasses the following datasets: Cleveland (303 observations), Hungarian (294 observations), Switzerland (123 observations), Long Beach VA (200 observations)¹⁹, and Statlog (Heart) Data Set (270 observations)²⁰. Initially totalling 1190 observations, the dataset was refined by removing 272 duplicates, resulting in a final dataset of 918 unique observations. For model evaluation, the data was split into training and testing sets with a ratio of 80:20, yielding 734 samples for training and 184 samples for testing.

Exploratory data analysis (EDA)

Figures 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 and 11 show the exploratory data analysis (EDA) of heart disease data revealing several critical insights. Heart disease can onset as early as 28 years old, with many cases occurring between 53 and 54 years. Both males and females predominantly develop heart disease around 54 to 55 years old, with males comprising 78.91% of the dataset and females 21.09%, making males 274.23% more prevalent than females. Geographically, Cleveland reports the highest number of cases (303), while Switzerland has the fewest (123).

Distribution of HeartDisease

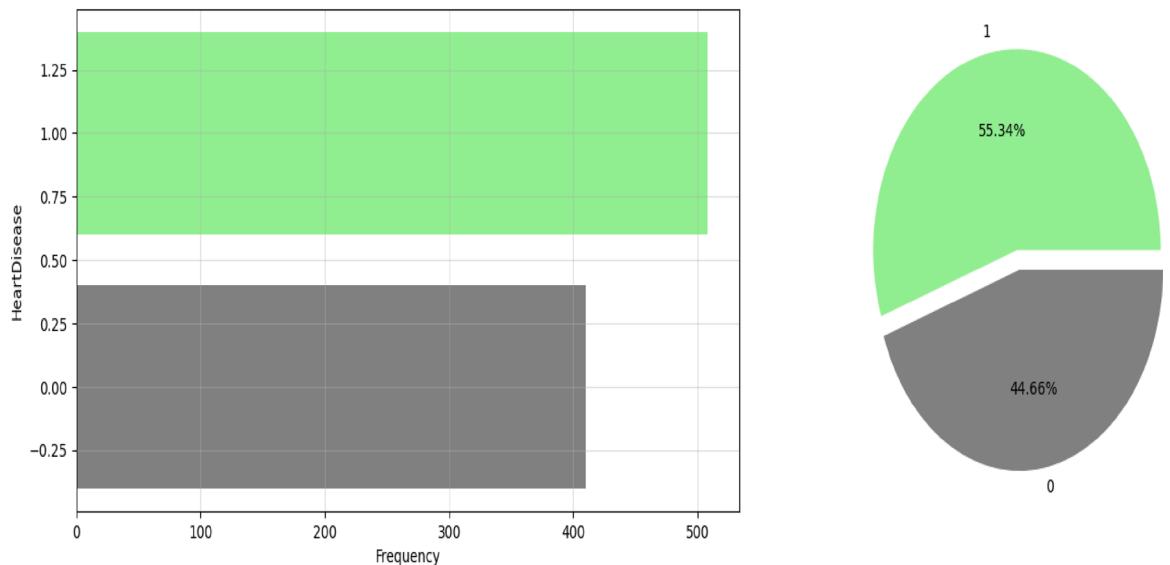


Fig. 1. Distribution of heart disease dataset where one represents heart disease and zero represents normal.

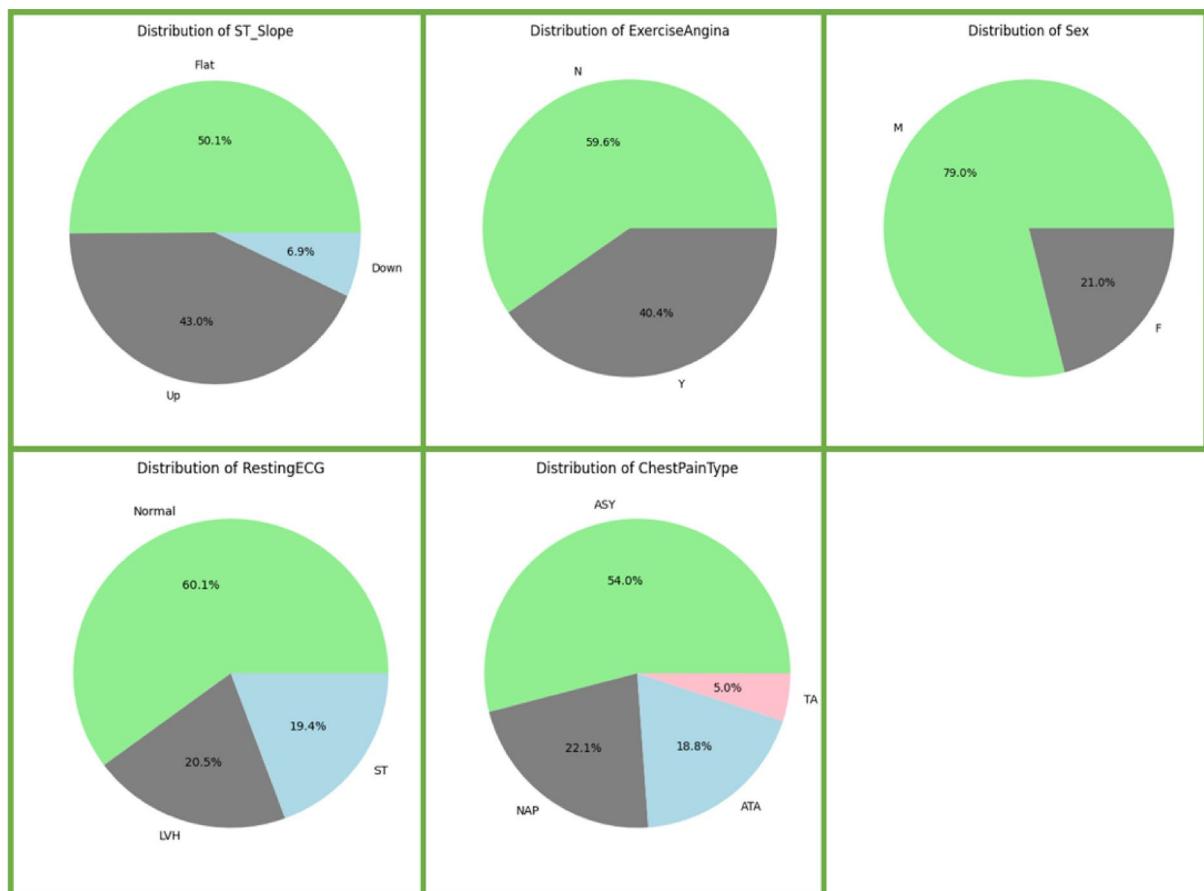
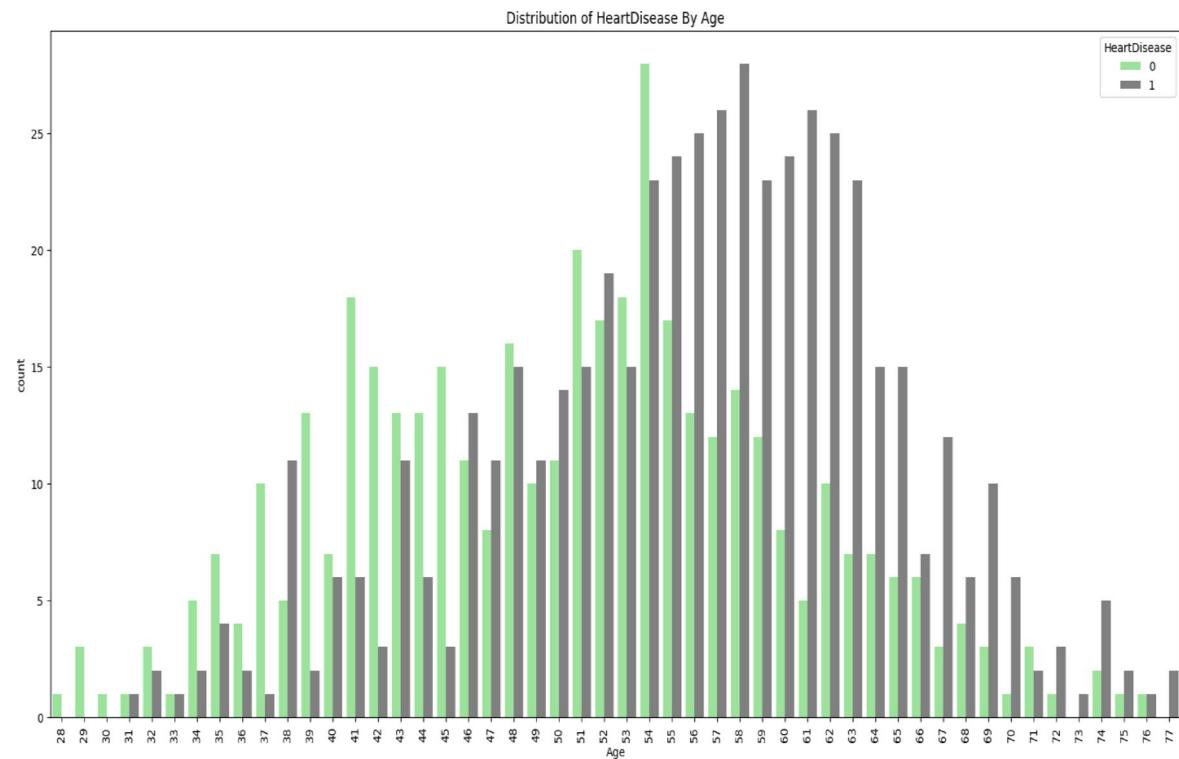
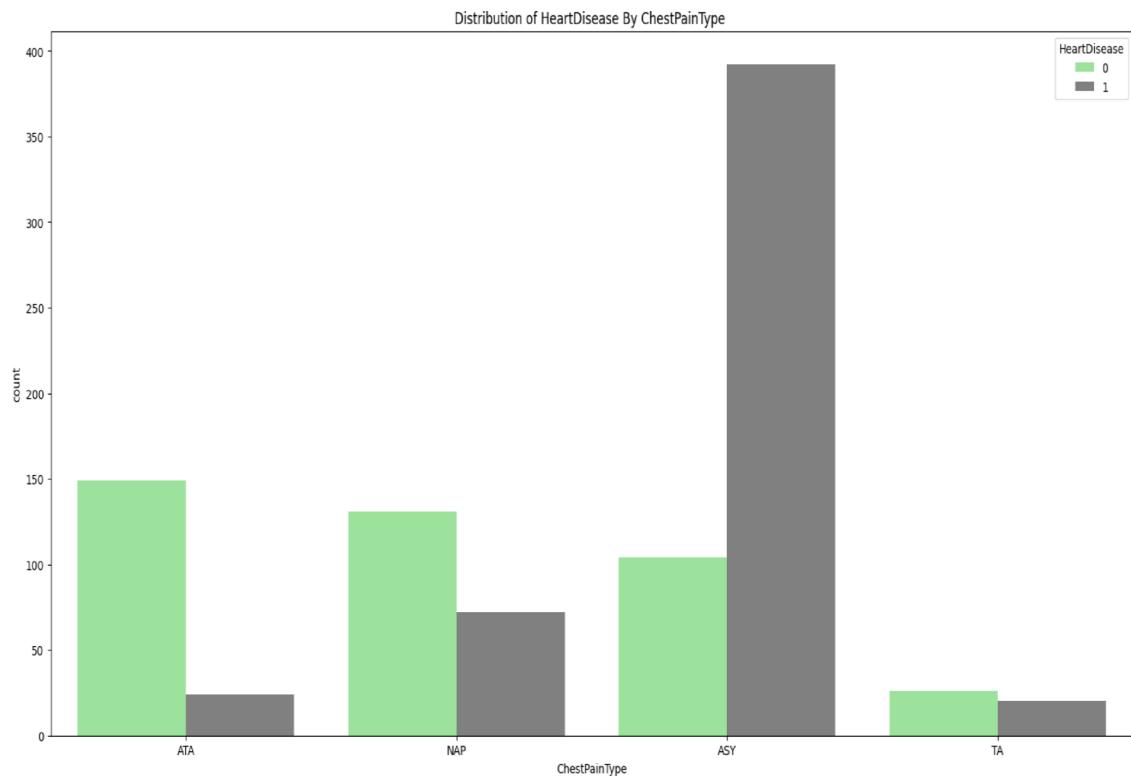


Fig. 2. Data distribution of various features.

**Fig. 3.** Distribution of heart disease dataset by age factor.**Fig. 4.** Distribution of heart disease dataset by chest pain types.

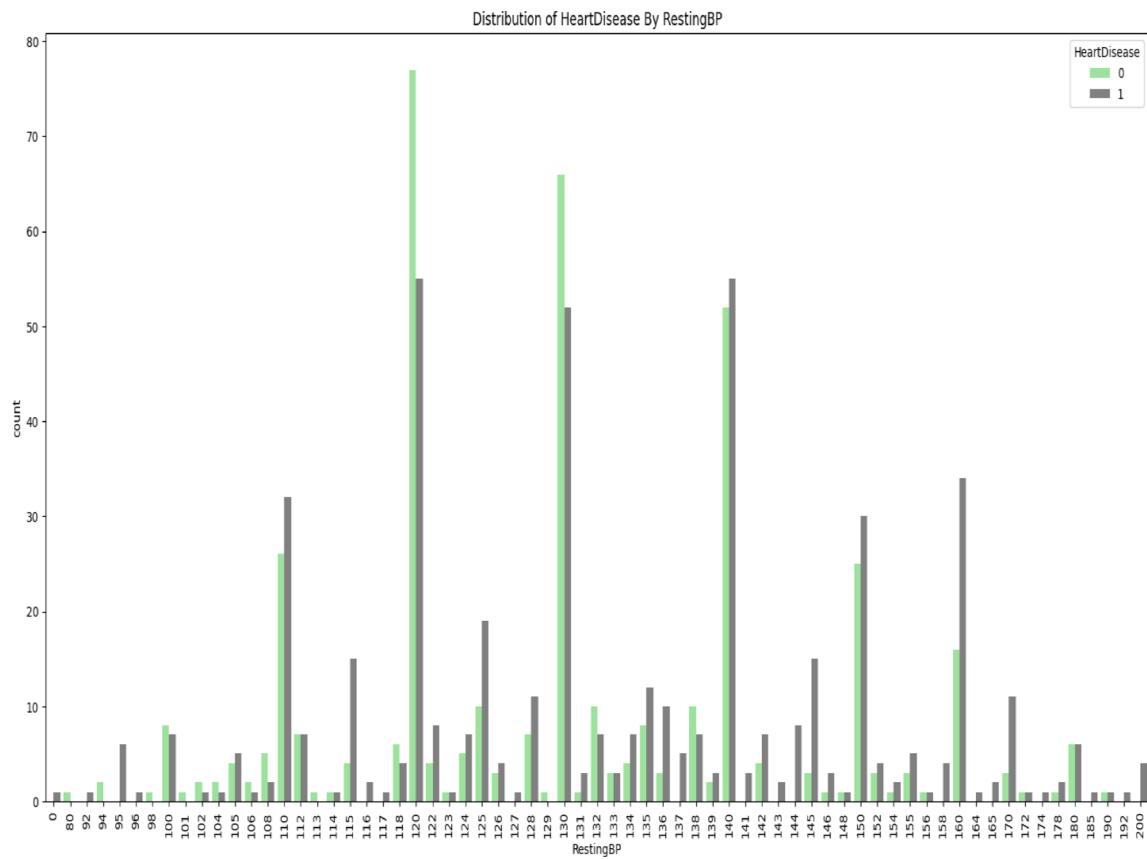


Fig. 5. Distribution of heart disease dataset by resting blood pressure factor.

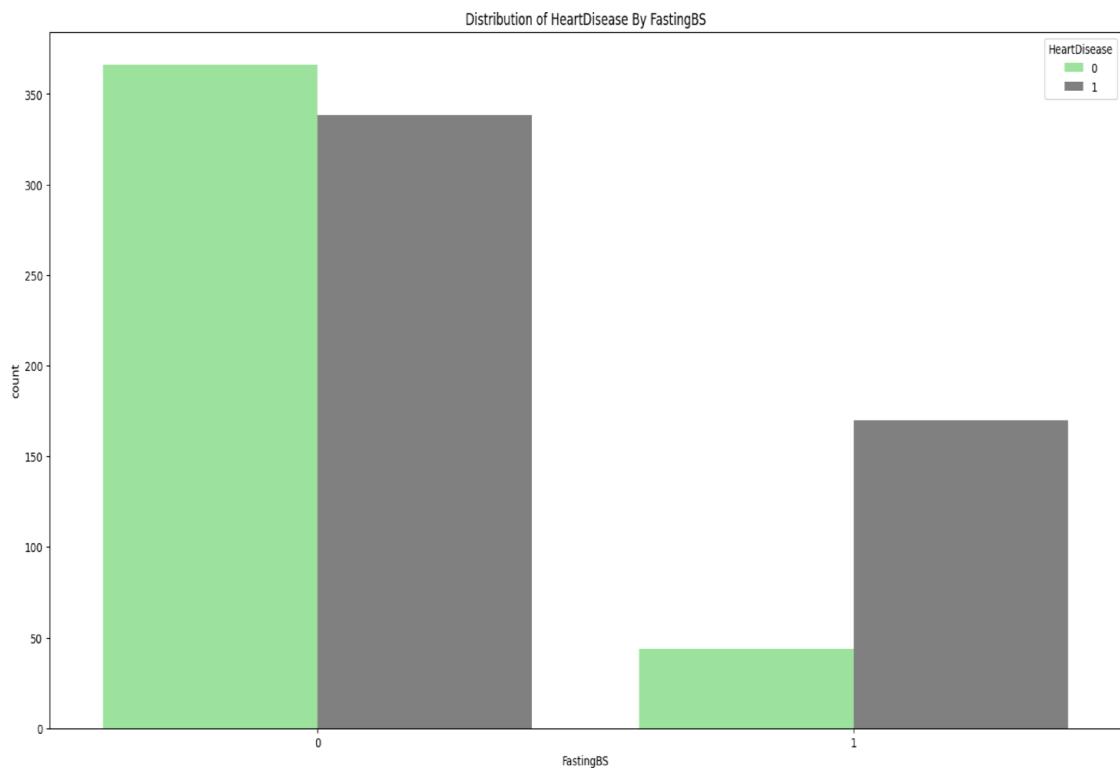


Fig. 6. Distribution of heart disease dataset by fasting blood sugar factor.

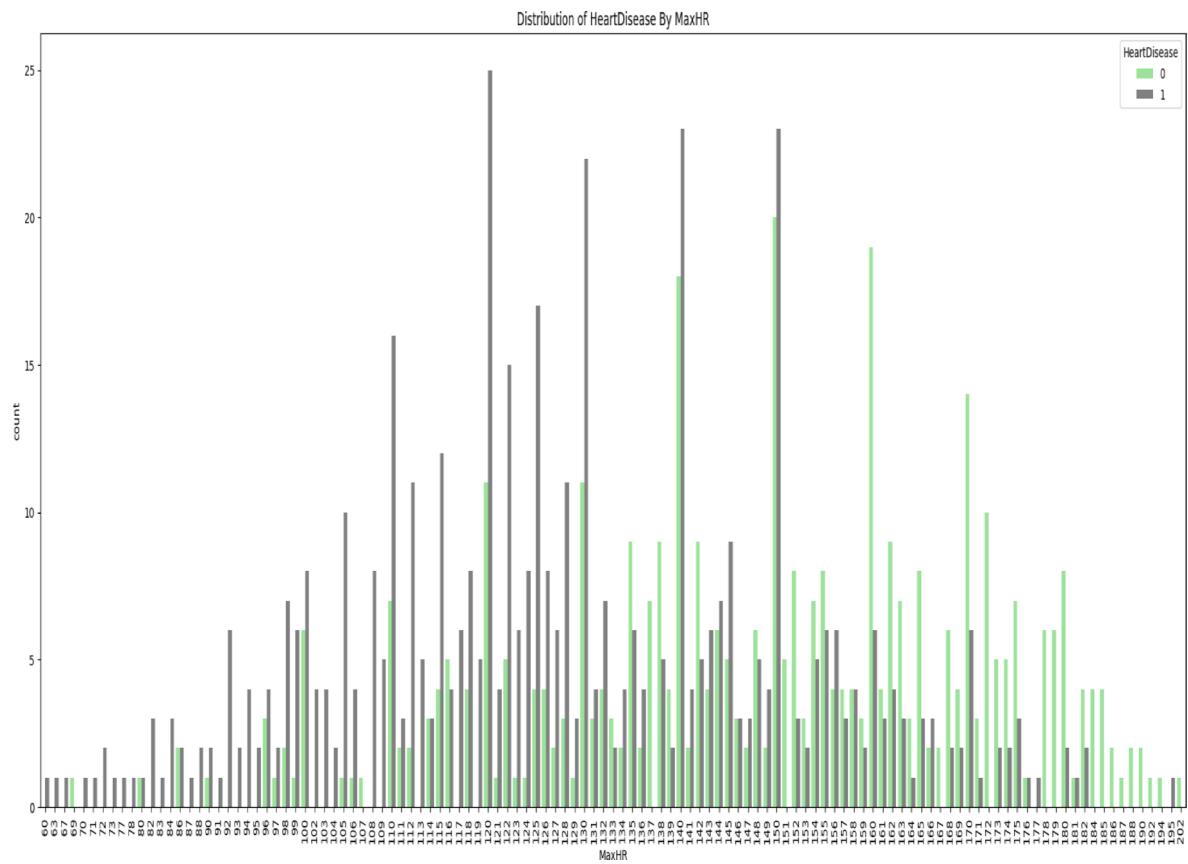


Fig. 7. Distribution of heart disease dataset by maximum heart-rate factor.

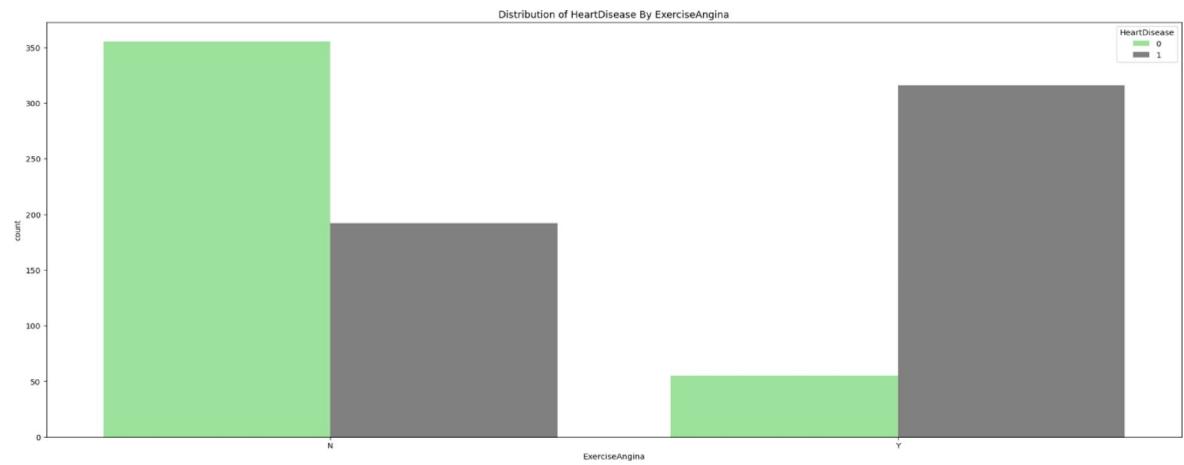


Fig. 8. Distribution of heart disease dataset by exercise angina factor.

Gender distribution by origin shows Cleveland having the highest number of females (97), and VA Long Beach the fewest (6). Conversely, Hungary has the highest number of males (212) and Switzerland the fewest (113). Chest pain analysis indicates that Cleveland leads in cases of Typical Angina, Asymptomatic, and Non-anginal chest pain, while Hungary has the most Atypical Angina cases. Switzerland reports the lowest for all chest pain types. Age-wise, Asymptomatic Angina peaks at ages 56–57, Typical Angina at 62–63, Non-anginal chest pain at 54–55, and Atypical Angina also at 54–55 years. These insights highlight the significance of age, gender, and geographical origin in understanding heart disease prevalence and characteristics. Figure 12 reveals the Pearson correlation between features and the target, ensuring the relevancy of each feature to the target variable. This correlation analysis helps identify which features have the most significant relationships with heart disease, guiding further analysis and model building.

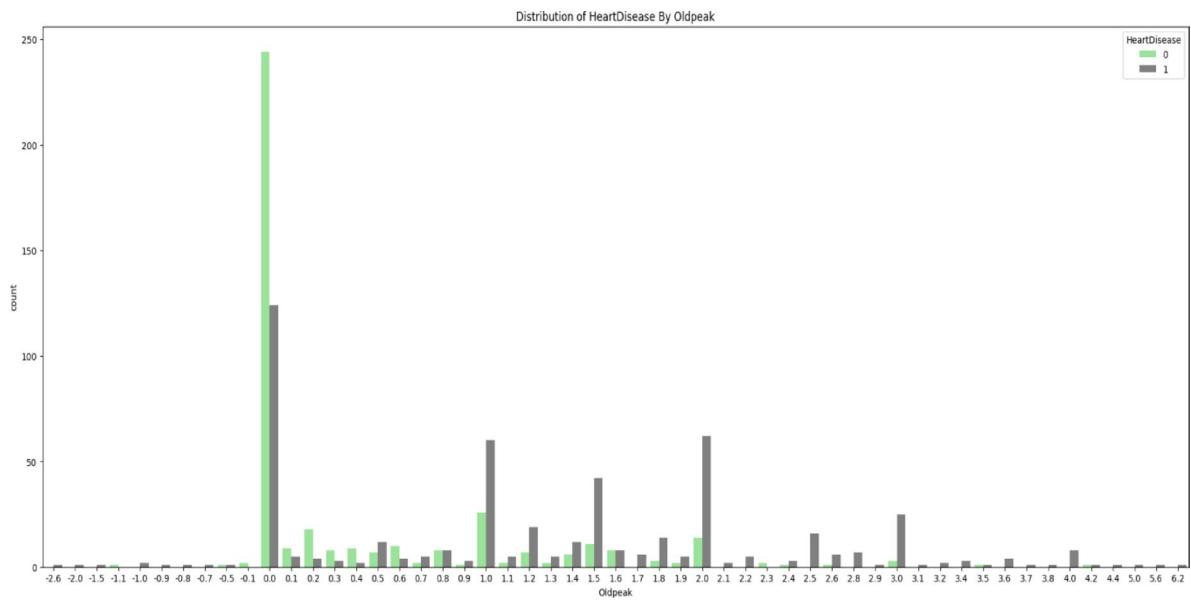


Fig. 9. Distribution of heart disease dataset by oldpeak factor.

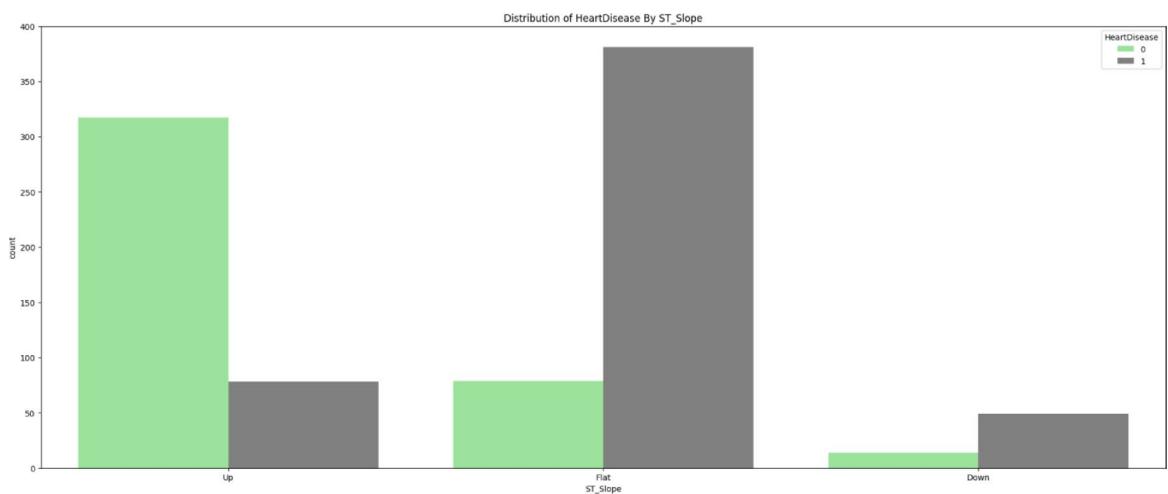


Fig. 10. Distribution of heart disease dataset by the slope of the peak exercise ST segment factor.

Methods

Classical support vector machine (CSVm)

CSVMs excel in separating data points belonging to two distinct classes. It achieves this by finding the optimal hyperplane in a high-dimensional feature space as shown in Fig. 13. This space is reached by mapping the original input vectors (x) using a function (Φ). Given a dataset with n samples²¹:

- Each sample is denoted as X_i, Y_i where:
 - X_i : An input vector with d dimensions.
 - Y_i : The class label associated with X_i .

CSVM aims to find a linear decision function that maximizes the margin between the two classes within this high-dimensional space. This decision function denoted as $f(x)$, is defined as:

$$f(x) = \langle w, \Phi(x) \rangle + b \quad (1)$$

- $\langle w, \Phi(x) \rangle$: Represents the dot product between a weight vector (w) and the mapped input vector ($\Phi(x)$)
- b : Bias term.

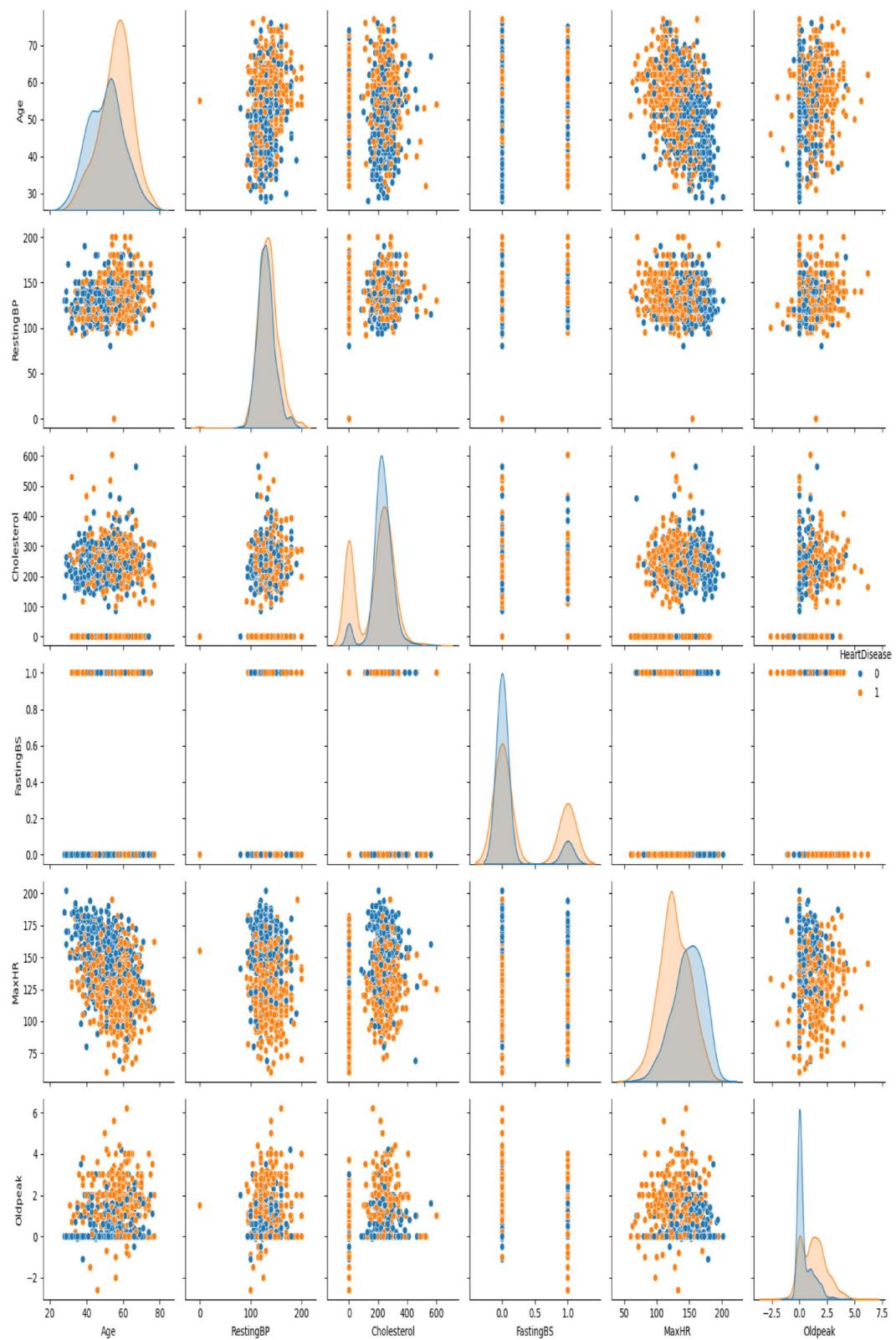


Fig. 11. Overall distribution of heart disease dataset with features.

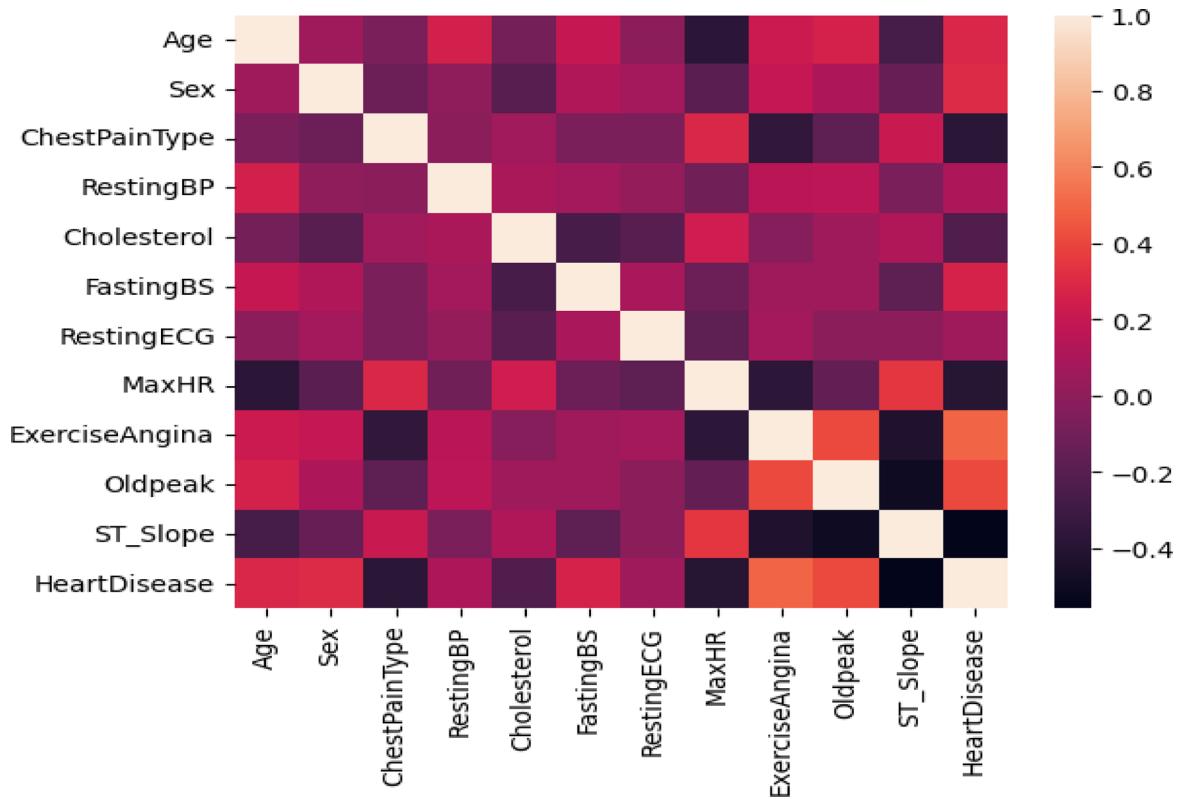


Fig. 12. Pearson correlation heatmap of features with respect to target.

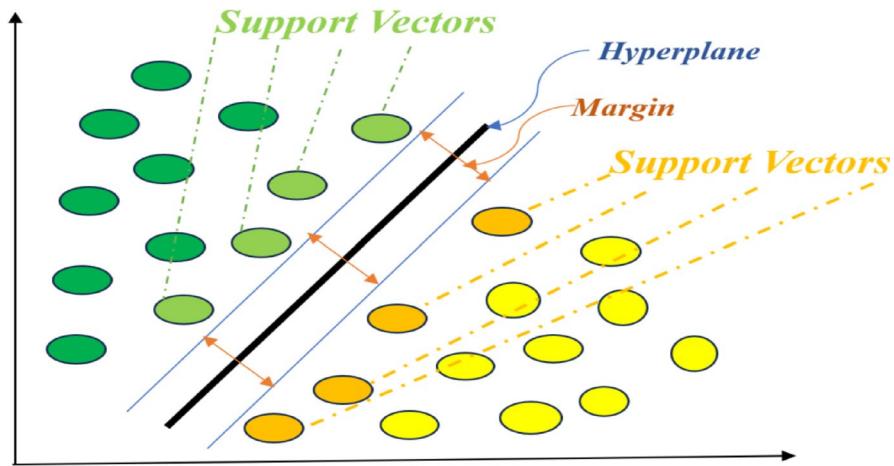


Fig. 13. SVM methodology.

The parameters w and b are determined by solving a convex quadratic optimization problem²¹.

Classical genetic algorithms (CGA)

Genetic algorithms (GAs) are a general adaptive optimization search methodology inspired by Darwinian natural selection and genetics in biological systems, offering a promising alternative to traditional heuristic methods. GAs operate with a population of candidate solutions. Following the principle of “survival of the fittest,” GAs iteratively refine these solutions to obtain an optimal result. Through successive generations of alternate solutions, each represented as a chromosome (a potential solution to the problem), GAs continue to evolve until acceptable results are achieved. Due to their combined exploitation and exploration search capabilities, GAs efficiently handle large search spaces and are less likely to settle on local optima compared to other algorithms²². A fitness function evaluates the quality of each solution during the assessment step. The crossover and mutation functions are the primary operators that randomly influence the fitness value. Chromosomes are chosen for

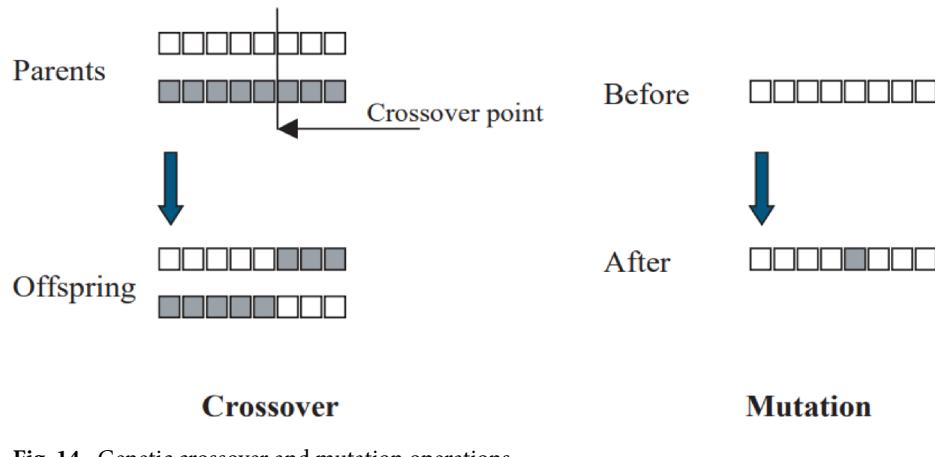


Fig. 14. Genetic crossover and mutation operations.

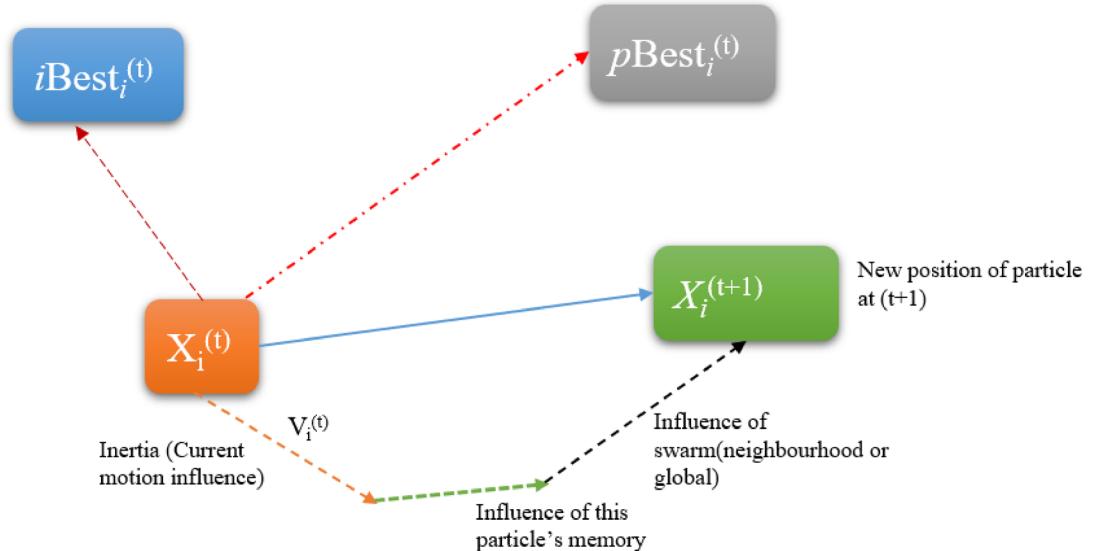


Fig. 15. CPSO methodology.

reproduction based on their fitness value, with fitter chromosomes having a higher probability of being selected for the recombination pool. This selection process typically uses methods such as roulette wheel or tournament selection²³.

Figure 14 illustrates the genetic operators of crossover and mutation. Crossover, a crucial genetic operator, enables the exploration of new solution regions in the search space by randomly exchanging genes between two chromosomes. This can be done using one-point crossover, two-point crossover, or homologous crossover. Mutation involves occasionally altering genes, such as changing a gene's code from 0 to 1 or vice versa in binary-coded genes.

Classical particle swarm optimization (CPSO)

Inspired by Nature's Collective Behaviour: Understanding CPSO. Swarm intelligence is a powerful approach to solving complex problems. Imagine a flock of birds effortlessly navigating through the sky. This seemingly simple behaviour emerges from the interactions between individual birds. CPSO developed in 1995, mimics this concept to tackle challenging optimization tasks²⁴. Core Idea, Collaboration towards the Best Solution, CPSO leverages the cooperation of numerous particles, each representing a potential solution as shown in Fig. 15. These particles move through a designated search space, constantly adjusting their trajectory based on two key pieces of information:

- Global Best: The best solution discovered so far by the entire swarm.
- Personal Best: Each particle's best-performing position encountered during its exploration.

This collaborative approach allows the swarm to progressively converge towards the optimal solution. The key Component of a CPSO Algorithm, *Search Space*: The defined area where particles explore potential solutions. *Objective Function*: A mathematical formula that determines the “goodness” of a solution. The goal is to find a solution that minimizes or maximizes this function (depending on the problem).

Essential Properties of a Particle.

- Position: Represents the particle’s coordinates within the search space.
- Velocity: Dictates the particle’s movement direction and magnitude.
- Memory: Each particle remembers:
 - Its personal best position.
 - The position of the swarm’s best neighbors (discovered by the entire group).

The Iterative Process of CPSO.

- Particle Movement: Based on its current position, velocity, and memory, each particle updates its position in the search space.
- Evaluation: The objective function is used to assess the quality of the new position.
- Update Memory: If the new position is better than the particle’s personal best, the memory is updated.
- Global Communication: The particle with the best position discovered so far (global best) is identified and shared with the entire swarm.
- Repeat: Steps 1–4 are repeated until a stopping criterion is met (e.g., a certain number of iterations or reaching a satisfactory solution).

CPSO in action Imagine a swarm of bees searching for the best source of nectar. Each bee represents a particle, and the nectar quantity signifies the objective function value. Through continuous exploration and information sharing, the swarm eventually locates the most abundant flower patch (optimal solution).

Quantum support vector machine (QSVM)

The quantum support vector machine (QSVM) method was implemented using classical simulation environments (Qiskit Aer simulator) with quantum-inspired circuits, where input data $x \in \Omega$ is mapped to a quantum state via the feature map circuit $U_\phi(x)$ (shown in panel c of Fig. 16), followed by processing this state with a short-depth quantum circuit $W(\theta)$ composed of multiple layers of parameterized gates (illustrated in panel b of Fig. 16). While the framework is designed for quantum hardware, the current results were obtained via classical simulations due to limitations in available quantum hardware (e.g., qubit coherence, gate fidelity). Future work will deploy the model on IBM Q or Rigetti systems. For binary classification, a binary measurement diagonal in the Z-basis is applied to the state $W(\theta)U_\phi(x) |0^n\rangle$ and repeated shots construct the empirical distribution $P_y(x)$. Labels are assigned based on this distribution and an optimized bias parameter (b). During training, the

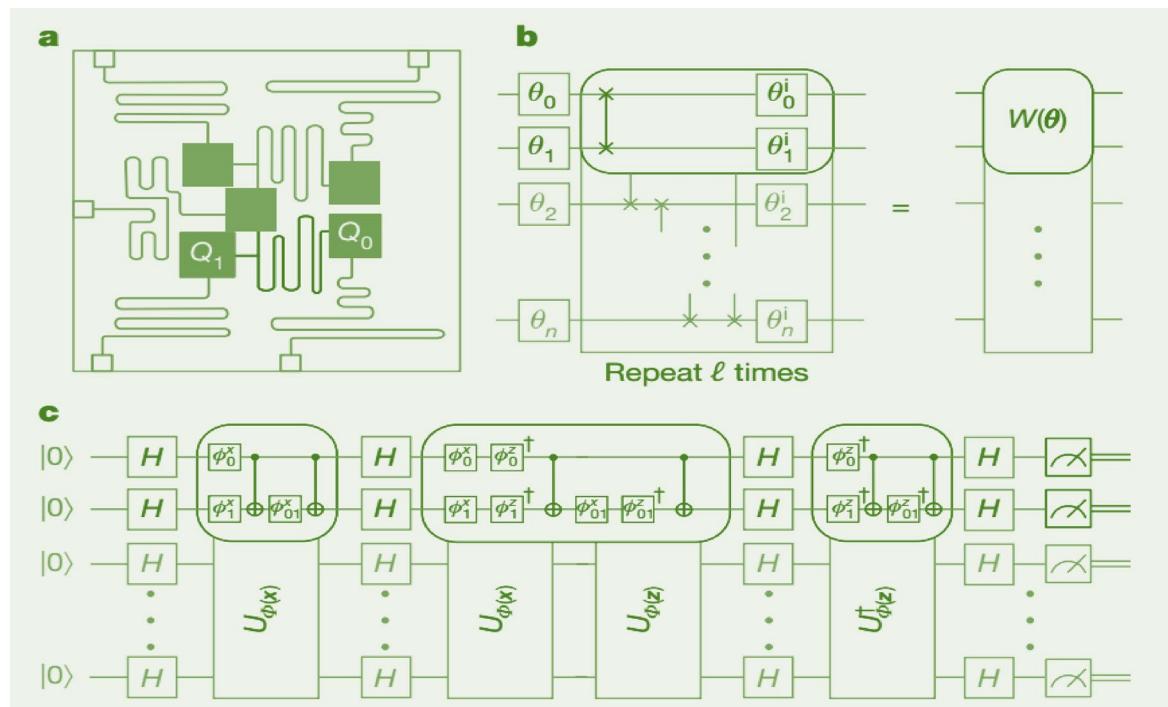


Fig. 16. Experimental implementation of QSVM.

parameters θ and (b) are optimized to minimize the empirical risk $R_{emp}(\theta)$ using the SPSA algorithm. Once the parameters converge to (θ^*, b^*) , the classifier assigns labels to unlabeled test data $s \in S$ using the same process. The image highlights the layout of the superconducting quantum processor (panel a), the structure of the variational circuit (panel b), and the feature map and measurement implementation (panel c), demonstrating the QSVM method's capability to classify data with higher accuracy as circuit depth increases²⁵.

Figure 16 shows the experimental implementation of QSVM. This figure provides a visual breakdown of three critical components of the QSVM methodology, each of which directly supports our proposed approach. The first panel (a) showcases the superconducting quantum processor used in this research experiments, highlighting the practical implementation of quantum circuits and their relevance to proposed quantum-inspired model (QGA-QPSO-QSVM). This hardware representation underscores the feasibility of applying quantum principles to real-world medical diagnostics, bridging the gap between theoretical quantum machine learning and its tangible execution. The second panel (b) illustrates the variational quantum circuit structure, which consists of parameterized quantum gates (e.g., rotation and entanglement gates) designed to map classical data into a high-dimensional quantum feature space. This aligns with our quantum feature map (Eq. 2 in Sect. 3.3.4), where input data is transformed into quantum states for enhanced classification. The third panel (c) details the feature map and measurement process, demonstrating how quantum kernel estimation is performed to compute the inner products of quantum states, a key step in QSVM classification. This visual aid directly connects to QSVM pseudocode (Table 3), clarifying how quantum states are manipulated and measured to generate predictions.

The relevance of Fig. 16 to the proposed work is multifaceted. First, it provides methodological clarity by illustrating how quantum circuits are constructed and executed, making our quantum ML approach more transparent and reproducible. Second, it validates performance by demonstrating that increasing quantum circuit depth improves classification accuracy, a finding that aligns with the proposed experiment results

Input:
<ul style="list-style-type: none"> • Labeled training samples $T = \{(\vec{x}_i, y_i) (\vec{x}_i \in \Omega \subset R^n, y_i \in C\}$ • Unlabeled test samples $S = \{\vec{s}\}$. • Number of measurement shots R • Initial parameters $\vec{\theta}_0$ and bias b
Training Phase:
<ol style="list-style-type: none"> 1: Initialize $\vec{\theta} = \vec{\theta}_0$. 2: while empirical risk $R_{emp}(\vec{\theta})$ not converged: 3: for each (\vec{x}_i, y_i) in T: 4: Initialize counters $r_y = 0 \forall y \in C$ 5: for shot = 1 to R: 6: Prepare state $\langle \Phi(\vec{x}_i)\rangle$ using $U_\phi(\vec{x}_i)$ 7: Apply discriminator circuit $W(\vec{\theta})$ 8: Measure outcome y using $\{M_y\}_{y \in C}$ 9: Increment y by setting $r_y \rightarrow r_y + 1$ 10: Compute empirical distribution $\hat{p}_y(\vec{x}_i) = r_y/R$ 11: Add error probability $Pr(\hat{m}(\vec{x}_i) \neq y_i m(\vec{x}_i) = y_i)$ with $\vec{p}_y(\vec{x}_i)$ and y_i to $R_{emp}(\vec{\theta})$ 12: Update $\vec{\theta}$ via optimization routine 13: return optimized $\vec{\theta}^*$
Classification Phase:
<ol style="list-style-type: none"> 1: Initialize counters $r_y = 0 \forall y \in C$ 2: for shot = 1 to R: 3: Prepare state $\langle \phi(\vec{s})\rangle$ using $U_\phi(\vec{s})$ 4: Apply $W(\vec{\theta}^*)$ 5: Measure outcome y 6: Increment r_y 7: return class label $\text{argmax}_y(r_y/R)$ $\text{argmax}_y(r_y/R)$

Table 3. Pseudocode of QSVM.

(Sect. 5.2), where QSVM achieved 97.83% accuracy compared to classical SVM's 94.02%. Third, it reinforces the quantum advantage in handling complex, non-linear relationships in medical data, which is critical for robust heart disease prediction.

QSVM is an advanced variant of the traditional Support Vector Machine (SVM) that leverages the principles of quantum mechanics, such as superposition and entanglement, to enhance computational capabilities. QSVM addresses the challenge of separating data that is not easily divided by a simple hyperplane in its original space by applying a non-linear transformation function, called a feature map, which projects the data into a higher-dimensional space known as feature space or Hilbert space. This transformation allows for the effective separation of the data. The pseudo-code for QSVM is represented in Table 3. The conversion of data into a quantum state is achieved through the feature map:

$$\Phi : \vec{x} \in \Omega \rightarrow |\Phi(\vec{x})\rangle \langle \Phi(\vec{x})| \quad (2)$$

Equation 2 represents the fundamental quantum feature mapping process that distinguishes QSVM from classical approaches. Where \vec{x} denotes a classical input vector (e.g., patient features such as age, cholesterol levels, etc.) and Ω represents the domain of possible input features. This equation describes the transformation of classical input data into a quantum state through a specifically designed quantum circuit. The feature map Φ serves as a bridge between classical data and quantum computation by encoding input features into the amplitudes of a quantum state. This quantum embedding enables the algorithm to operate in an exponentially larger Hilbert space compared to classical methods, allowing for more efficient representation of complex relationships in the data. This mapping is implemented via a parameterized quantum circuit (PQC), where input features determine rotation angles of quantum gates (e.g., $R_Y(x_i)$). The notation $|\Phi(\vec{x})\rangle \langle \Phi(\vec{x})|$ represents the density matrix formulation of the quantum state, which captures both the superposition of states and quantum entanglement properties. In practical implementation, this mapping is achieved through parameterized quantum gates (like rotation and entanglement gates) that transform the initial state according to the input features. The significance of this quantum feature map lies in its ability to implicitly compute high-dimensional inner products (kernel evaluations) that would be computationally prohibitive for classical systems. This quantum advantage directly contributes to the enhanced classification performance demonstrated in our results (Sect. 5.2), where QSVM achieved 97.83% accuracy compared to classical SVM's 94.02%. The feature map's design also influences the model's capacity to handle non-linear decision boundaries, as the quantum kernel can capture complex patterns that classical kernels might miss.

The QSVM algorithm involves computing the inner product for each pair of data points in the set to evaluate their similarity, which is then used to classify the data points within this transformed feature space. The collection of these inner products forms the kernel matrix. Figure 17 provides an overview of QSVM classification. The implementation of QSVM consists of three main steps:

1. Preprocessing: This phase includes standard data preparation techniques such as scaling, normalization, and Genetic algorithms.
2. Generation of the Kernel Matrix: This step involves computing the kernel matrix based on the inner products of the data points in the feature space.
3. QSVM Classification: The final step is the classification process, which involves estimating the kernel for a new set of test data points and using this information to classify them accordingly.

Quantum genetic algorithms (QGA)

The genetic algorithm (GA) is a global optimization method renowned for its robustness and versatility, capable of addressing complex problems that traditional optimization methods cannot solve. Recently, quantum genetic algorithms (QGAs), which leverage principles from quantum computing, have gained significant attention. The QGA leverages quantum-inspired operations (e.g., qubit superposition, quantum gates) simulated on classical hardware to explore the feature space. The algorithm initializes qubit chromosomes (allowing any linear superposition of solutions) with probability amplitudes (α_i, β_i) and updates them via quantum rotation gates. This results in rapid convergence, strong global search capabilities, simultaneous exploration and exploitation, and small population sizes without performance degradation. Though the current implementation uses classical simulations (Python/Qiskit), the formulation is compatible with quantum hardware for future scalability. QGAs use a qubit chromosome representation by probability amplitudes (e.g., $\alpha|0\rangle + \beta|1\rangle$), allowing any linear superposition of solutions. Unlike conventional GAs that rely on crossover and mutation to maintain population diversity, QGAs update probability amplitudes of quantum states using quantum logic gates. The Pauli-X gate (bit-flip) introduces diversity by inverting qubit states (e.g., $|0\rangle \leftrightarrow |1\rangle$), mimicking classical mutation. The Hadamard gate initializes qubits into superposition ($|0\rangle \rightarrow |0\rangle + |1\rangle$), allowing parallel evaluation of solutions. During evolution, the rotation gate ($R\theta$) fine-tunes amplitudes by applying a phase shift θ , guided by fitness feedback to steer the population toward optimal solutions. For crossover operations, the CNOT gate entangles qubits, preserving trait correlations by flipping a target qubit conditionally ($|00\rangle \rightarrow |00\rangle, |10\rangle \rightarrow |11\rangle$, $|00\rangle, |10\rangle \rightarrow |11\rangle$). Together, these gates enable QGAs to exploit quantum parallelism and entanglement, achieving faster convergence and higher accuracy than classical counterparts. For instance, in feature selection (Table 4), rotation gates adaptively prioritize high-impact features, while CNOT gates maintain synergistic feature pairs, reducing redundancy. This gate-based workflow underpins the QGA's ability to solve complex problems like heart disease prediction with fewer evaluations and smaller populations. This results in rapid convergence, strong global search capabilities, simultaneous exploration and exploitation, and small population sizes without performance degradation. QGA is applied to a combinatorial optimization problem. The process begins with initializing the population and qubits, constructing observation states from probability amplitudes, and evaluating fitness using a defined function. The optimal individual is maintained, and if termination conditions

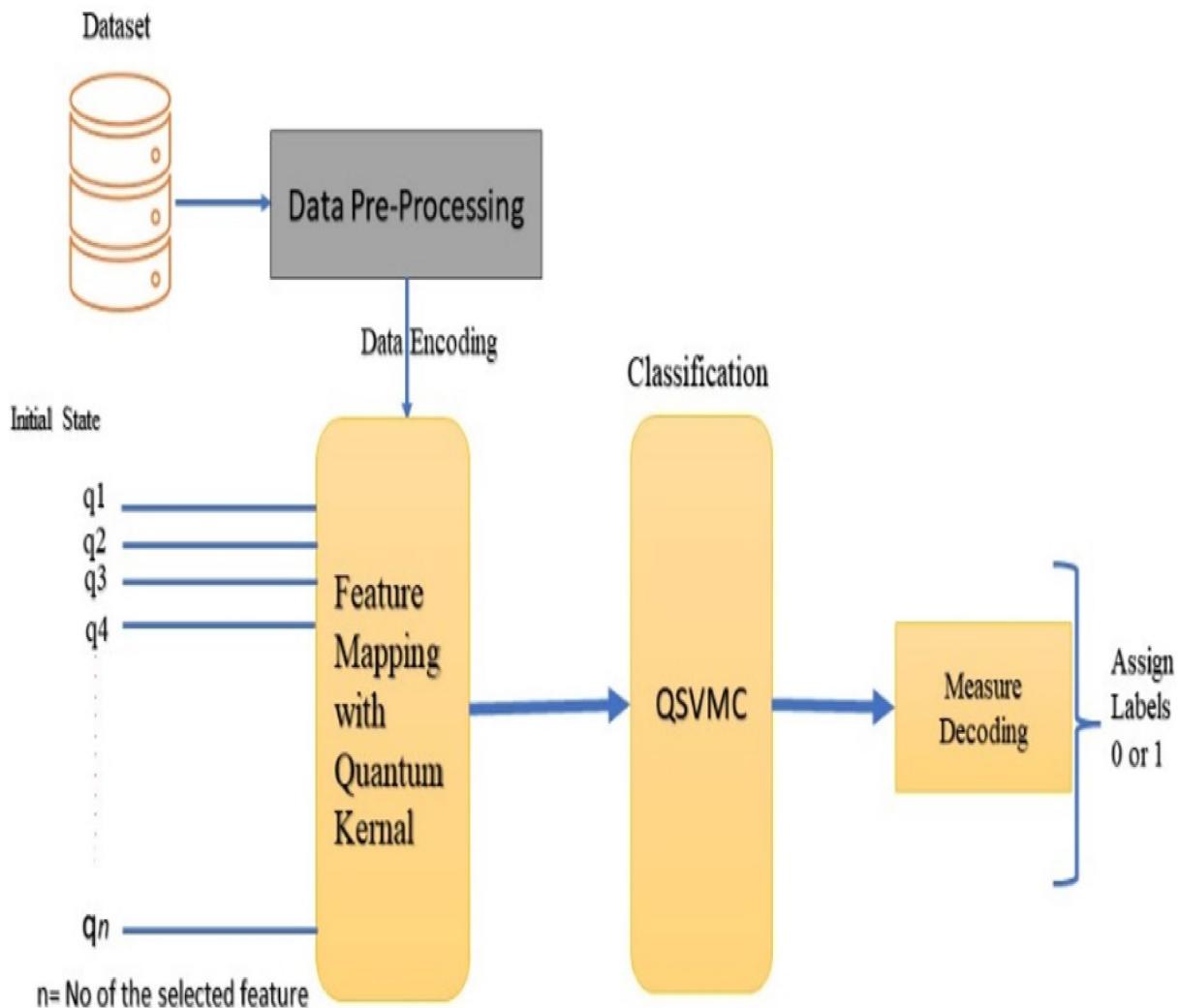


Fig. 17. QSVM classification methodology.

are met, the algorithm stops. Otherwise, quantum rotation gates update the probability amplitudes, and the process repeats until the optimal solution is found. The QGA is employed to automatically search for the optimal feature subset from the original feature set. The steps are as follows:

1. Initialization involves selecting the population size (h) and the number (m) of qubits. The population (P) comprises (h) individuals (p_1, p_2, \dots, p_h), with everyone P_j represented by a matrix of probability amplitudes α_i and β_i for the quantum states 1 and 0, respectively. Initially, α_i and β_i are set to $\frac{1}{\sqrt{2}}$ for all qubits, indicating equal superposition. The evolutionary generation counter starts at 0.

2. The observation state (R) is constructed from the probability amplitudes, resulting in a binary string for everyone.

3. The fitness function evaluates individuals in population (P). For a feature vector of dimension (d) and a binary string (b_1, b_2, \dots, b_h) the fitness function (f) aims to minimize the class separability reduction matrix. The optimal individual is maintained, and the algorithm checks for termination. If the termination condition is satisfied, the algorithm ends; otherwise, it continues. Quantum rotation angles are determined and applied using quantum rotation gates to update probability amplitudes. The generation counter (gen) is incremented, and the process repeats from step 2 as shown in Table 4 pseudocode.

Quantum particle swarm optimization (QPSO)

Physics, a cornerstone of modern science and technology, offers profound insights into the natural world. Inspired by both biological evolution and the power of simulation, scientists have developed numerous successful theories. The Simulated Annealing Algorithm (SAA) exemplifies this fruitful interplay of ideas. QPSO takes this approach a step further. It leverages the concept of quantum bits (qubits) to define particles within the swarm. QPSO employs quantum particles simulated classically to navigate the hyperparameter space. Each particle's position is encoded as a qubit state ($q(i)^{(j)(t)}$), and updates follow quantum-inspired rules (e.g., tunnelling

```

Procedure QGA ():

# Step 1: Initialization
Population P ← Initialize with quantum states
gen ← 0

# Step 2: Iterative optimization
while not Termination Condition ():

# Step 3: Construct observation state (R)
R ← Decode quantum states in P to binary strings

# Step 4: Evaluate fitness
Fitness f ← Evaluate (P)

# Step 5: Update optimal individual
Update Best (P)
if Termination Condition ():
    break

# Step 6: Apply quantum gates
Apply Quantum Rotation (P)
gen ← gen + 1

return Best Solution (P)
End Procedure

```

Table 4. Pseudocode of QGA.

effects). The simulations were performed using Qiskit's statevector simulator, with plans to test on IBM Q's superconducting qubits in subsequent work. Unlike traditional PSO, QPSO employs a random observation process instead of a sigmoid function. Additionally, it incorporates guidance from the best-performing chromosome to progressively converge towards the optimal solution. This integration of quantum mechanics and swarm intelligence principles distinguishes QPSO as a novel and potentially powerful approach to discrete optimization problems. Figure 18 represents the flowchart of QPSO and Table 5 represents the pseudo code of QPSO. In QPSO, a fundamental concept is the quantum particle vector. This vector, denoted as $a^{(B)}$, represents a single particle within the swarm and utilizes qubits to encode information. Unlike classical bits restricted to 0 or 1, qubits can exist in a superposition of both states simultaneously. The $a^{(B)}$ vector is defined with the following properties:

- $q(i)^{(j)(t)}$: This represents the probability of the i -th bit in the j -th particle being in state 0 at the t -th generation.
- m : This signifies the length of the particle, essentially the number of bits it carries.
- N : This represents the total population size, indicating the number of particles within the swarm.

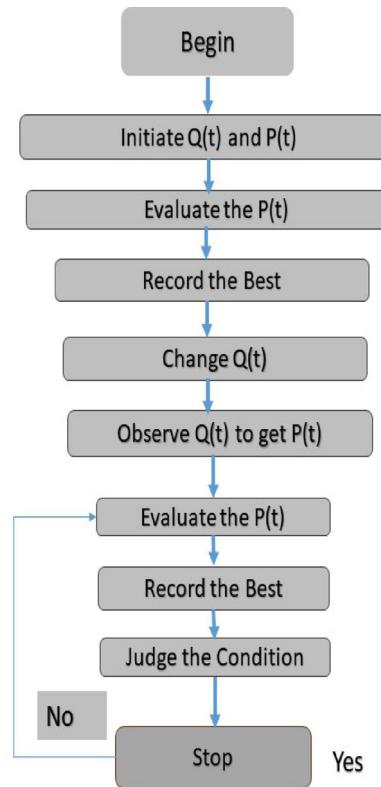
In QPSO, the fitness value of each particle is calculated based on the classification performance of the QSVM model when trained with the hyperparameters represented by that particle. The fitness function evaluates the model's accuracy, which is defined as the ratio of correctly classified instances to the total number of instances in the dataset. Mathematically, the fitness function for the j th particle at iteration t is computed as:

$$\text{Fitness} (j, t) = \frac{\text{Number of Correctly Classified Instances}}{\text{Total Number of Instances}} \quad (3)$$

Additionally, to enhance optimization, we incorporate a penalty term for model complexity, ensuring that the selected hyperparameters do not lead to overfitting. The modified fitness function is given by:

$$\text{Fitness} (j, t) = \text{Accuracy} - \lambda \cdot (\text{Model Complexity}) \quad (4)$$

Here, λ is a that balances accuracy and complexity. The model complexity can be quantified based on the number of support vectors in QSVM or the magnitude of hyperparameters. During the optimization process, each particle's position corresponds to a candidate set of hyperparameters (e.g., kernel parameters, regularization

**Fig. 18.** QPSO flowchart.

```

# Parameters
populationsize = N
maxiterations = M

# Initialize swarm
particles ← Initialize Quantum Particles (N)

for iteration = 1 to M:
    for each particle in particles:
        # Update position via quantum behaviour
        particle.position ← Quantum Update (particle, globalbest)
    fitness ← Evaluate (particle)

    # Evaluate fitness
    if fitness > particle.bestfitness:
        particle.bestposition ← particle.position
        particle.bestfitness ← fitness

    if fitness > globalbest.fitness:
        globalbest ← particle

return globalbest
  
```

Table 5. Pseudocode of QPSO.

terms). The fitness value guides the swarm toward optimal hyperparameters by updating particle velocities and positions iteratively.

Feature selection for clinical interpretability

The feature selection process is designed to enhance interpretability and clinical utility. Using CGA and QGA, the model identifies high-impact features like exercise-induced angina and ST depression (Oldpeak), which are well-documented risk factors in cardiology. By reducing dimensionality (from 11 to 7–9 features), the model minimizes noise and focuses on actionable biomarkers, making it easier for clinicians to understand and validate predictions. Additionally, quantum-inspired optimization (QPSO) ensures that selected features are not only statistically optimal but also clinically meaningful, as confirmed by correlation analysis (Fig. 12). This balance between accuracy and simplicity is crucial for real-world deployment, where speed and interpretability are paramount.

Performance metrics

To comprehensively evaluate model performance, various key metrics are employed as detailed in Table 6. Accuracy assesses overall prediction correctness, while the F1-score balances precision and recall. Precision quantifies correct identifications, and sensitivity (recall) evaluates the model's accuracy in identifying true positives. Specificity measures the model's ability to correctly identify true negatives (TN). Positive Likelihood Ratios (LR+) and Negative Likelihood Ratios (LR-) provide insights into the balance between true positive (TP) and false positive (FP) rates, while Diagnostic Odds Ratio (DOR) offers a ratio comparing the odds of a positive test result between subjects with and without the condition. Each metric contributes unique perspectives to understanding model performance across different aspects of classification and prediction tasks²⁶.

K-fold cross-validation

To validate the proposed models, tenfold cross-validation is employed on the initial dataset, dividing it into ten mutually exclusive subsets: (D1, D2, D3, D4, D5, D6, D7, D8, D9) and (D10). Each iteration of this process involves using one subset as the test set while the remaining nine subsets form the training set. This systematic rotation ensures that every subset is used exactly once as the test set, thereby reducing the potential for bias, and providing a robust evaluation of the model's performance. The accuracy of the model is computed as the ratio of correctly classified instances to the total number of instances in the dataset and is averaged across all ten iterations. This averaged accuracy metric serves as a reliable indicator of the model's classification efficacy across different subsets of data, offering valuable insights into its generalizability and performance compared to other methodologies or model variations. Tenfold cross-validation is particularly advantageous in providing a comprehensive assessment due to its balance between computational efficiency and reliable estimation of model performance²⁷.

Proposed methodology

The workflow illustrated in Fig. 19 outlines the proposed approaches using classical and quantum techniques.

The classical model (CGA-CPSO-CSVM)

The classical (CGA-CPSO-CSVM) model is a sophisticated hybrid approach that combines a CGA for feature selection, CPSO for hyperparameter tuning, and CSVM for classification to enhance predictive performance. The model was trained on 80% of the data (734 samples) and validated on the remaining 20% (184 samples) to ensure robust performance estimation. CGA begins by identifying the most relevant subset of features through iterative genetic operations, thereby reducing dimensionality, and improving computational efficiency. CPSO then optimizes the SVM's hyperparameters by navigating the hyperparameter space with a swarm of particles that iteratively update based on individual and collective best positions, aiming to enhance classification

S. No	Performance Metrics	Formula
1	Accuracy	$\frac{(TN+TP)}{(FP+TP+FN+TN)}$
2	F1-score	$\frac{2TP}{(2TP+FP+FN)}$
3	Precision	$\frac{TP}{(TP+FP)}$
4	Sensitivity (Recall/TPR)	$\frac{TP}{(TP+FN)}$
4	Specificity	$\frac{TN}{TN+FP}$
5	Matthews Correlation Coefficient (MCC)	$\frac{(TN \times TP) - (FP \times FN)}{\sqrt{(TN+FN)(TP+FP)(TN+FP)(FN+TP)}}$
6	Positive Likelihood Ratios (LR+)	$\frac{TPR}{FPR}$
7	Negative Likelihood Ratios (LR-)	$\frac{FNR}{TNR}$
8	Diagnostic Odds Ratio (DOR)	$\frac{LR+}{LR-}$

Table 6. Performance metrics.

Hybrid Approach using CGA/QGA with CPSO/QPSO

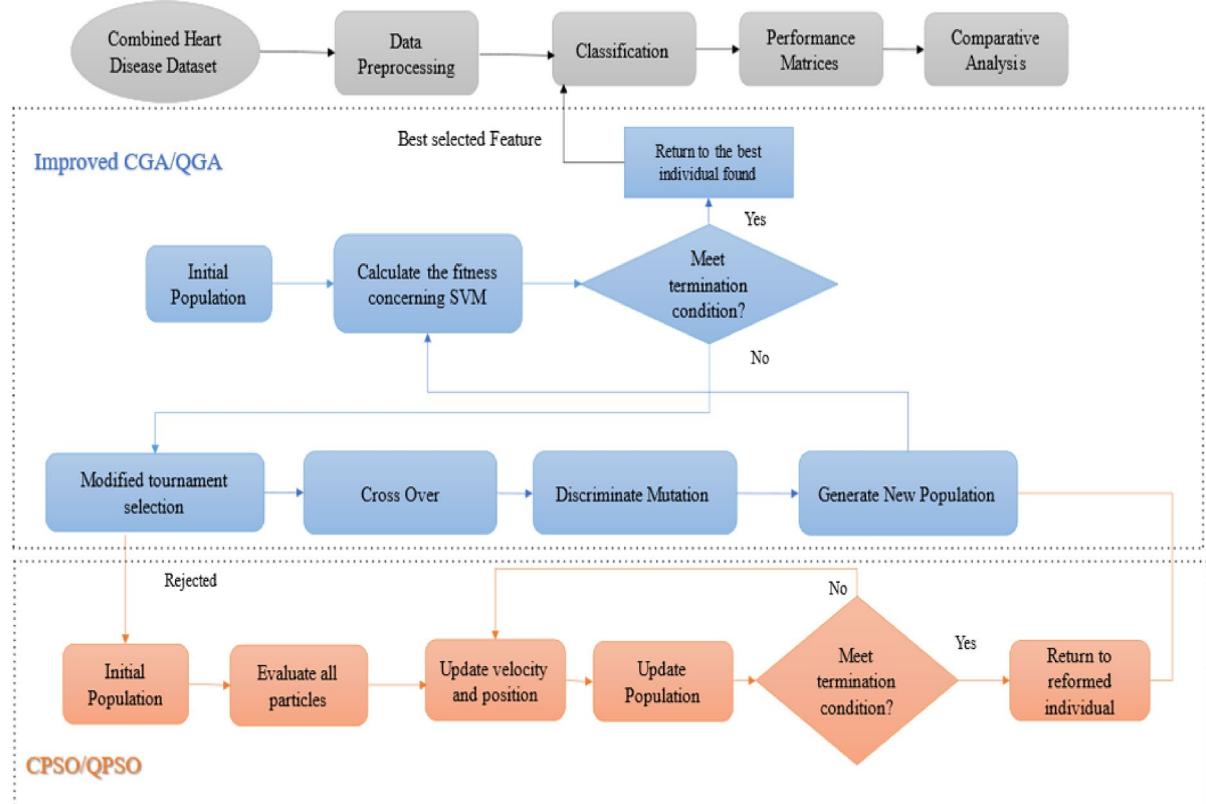


Fig. 19. Workflow of CGA/QGA with CSPO/QPSO with CSVM/QSVM.

accuracy. Finally, the CSVM, trained on the selected features and optimized hyperparameters, constructs a hyperplane that maximally separates the classes, resulting in improved classification accuracy and robustness. The exploratory Process of the classical Model has been outlined in Table 7 pseudocode.

The quantum model (QSVM-QGA-QPSO)

The Quantum (QSVM-QGA-QPSO) model is an advanced hybrid approach that combines QSVM for classification, QGA for feature selection, and QPSO for hyperparameter tuning to enhance predictive performance. In contrast to classical methods, QGA uses quantum-inspired operations to explore the feature space more efficiently, leveraging phenomena like superposition and entanglement to handle larger and more complex datasets. QPSO optimizes the QSVM's hyperparameters by navigating the hyperparameter space with quantum particles that follow quantum behaviour, such as tunnelling and coherence, to avoid local minima and find the global optimum more effectively. QSVM, unlike its classical counterpart, operates in a quantum-enhanced feature space, allowing it to solve complex classification problems that are infeasible for classical SVMs.

Quantum genetic algorithm (QGA) operations The QGA employs quantum-inspired genetic operations to enhance feature selection.

1. *Quantum Crossover:* Unlike classical crossover, which exchanges genetic material between two parents, quantum crossover operates on qubit chromosomes. It uses quantum gates (e.g., controlled-NOT gates) to entangle qubits from parent chromosomes, creating offspring with superpositions of states. This enables exploration of a broader solution space while preserving diversity.
2. *Quantum Mutation:* This operation introduces randomness by applying quantum rotation gates to adjust the probability amplitudes of qubits. A small rotation angle is randomly selected, and the qubit state is updated as $\begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} \alpha \\ \beta \end{bmatrix}$, where α and β are the probability amplitudes of the qubit states $|0\rangle$ and $|1\rangle$, respectively. This ensures controlled exploration without premature convergence.
3. *Quantum Selection:* Inspired by quantum measurement, this step collapses the superposition of qubit chromosomes into classical binary strings based on their probability amplitudes. Fitter solutions (higher probability amplitudes) are more likely to be selected, mimicking the survival-of-the-fittest principle.
4. *Decode Quantum Bits:* This function converts qubit chromosomes into classical feature subsets by measuring each qubit. For example, a qubit with amplitudes $\alpha=0.6$ and $\beta=0.8$ has a higher probability of collapsing to $|1\rangle$, indicating the corresponding feature is selected.

```

# ===== Feature Selection using CGA =====

Input:
  Dataset D with features F = {f1, f2, ..., fn}
  Population size N, Generations G, Crossover rate Pc, Mutation rate Pm

1: Initialize population P with N chromosomes (binary encoded feature subsets)
2: for generation = 1 to G:
3:   Evaluate fitness of each chromosome in P:
4:   Train CSVM with selected features → Accuracy → Fitness value
5:   Select parents via roulette wheel selection
6:   Apply one-point crossover (rate Pc) to create offspring
7:   Apply bit-flip mutation (rate Pm) to offspring
8:   Replace least-fit chromosomes in P with offspring
9:   if Convergence Criteria ()�:
10:    break
11: BestFeatures ← Chromosome with highest fitness

# ===== Hyperparameter Tuning using CPSO =====

Input:
  CSVM hyperparameters (e.g., C, γ)
  Swarm size S, Iterations I, Inertia weight w

12: Initialize particles X = {x1, x2, ..., xS} with random positions (C, γ)
13: for iteration = 1 to I:
14:   for each particle xi:
15:     Train CSVM with BestFeatures and xi's hyperparameters
16:     Evaluate fitness (e.g., validation accuracy)
17:     Update personal best (pBest) if current fitness > pBest
18:     Update global best (gBest) across swarm
19:   for each particle xi:
20:     Update velocity: vi = w · vi + c1 · rand() · (pBest - xi) + c2 · rand() · (gBest - xi)
21:     Update position: xi = xi + vi
22: BestHyperparams ← gBest

# ===== Classification using CSVM =====

23: Train CSVM with BestFeatures and BestHyperparams
24: Evaluate on test set → Accuracy, Precision, Recall, etc.

```

Table 7. Pseudocode for classical (CGA-CPSO-CSVM).

Quantum particle swarm optimization (QPSO) operations QPSO integrates quantum principles into PSO.

- *Quantum Particles*: Each particle's position is represented by a qubit string, enabling superposition of states. The velocity update incorporates quantum interference, where particles adjust their trajectories based on quantum potential wells centered on personal and global best positions.
- *Quantum Measurement*: Particles are observed (measured) to collapse into classical positions for fitness evaluation. The measurement probability is derived from the squared magnitude of qubit amplitudes ($|\alpha|^2$ and $|\beta|^2$).

QSVM classification The QSVM leverages a quantum feature map $U_\Phi(x)$ to project data into a high-dimensional Hilbert space. The kernel matrix is computed using quantum circuits that evaluate inner products via quantum parallelism. Training involves optimizing a quantum variational circuit $W(\theta)$ to find the optimal separating hyperplane.

```

// Quantum Genetic Algorithm (QGA) for Feature Selection
1. function QGA_Feature_Selection(data, population_size, generations, qubit_length):
    // Initialize population with quantum bits representing feature subsets
2.     Initialize population of chromosomes each with qubit_length quantum bits
3.     for generation in range(generations):
        // Evaluate fitness of each individual in the population
4.         for chromosome in population:
5.             Decode quantum bits to feature subset
6.             fitness = EvaluateFitness(chromosome, data)
7.             Store fitness value
        // Quantum-inspired selection
8.             selected_population = QuantumSelection(population, fitness)
        // Apply quantum crossover and mutation operations
9.             offspring_population = []
10.            for i in range(population_size // 2):
11.                parent1, parent2 = Quantum Crossover(selected_population)
12.                child1, child2 = Quantum Mutation(parent1), Quantum Mutation(parent2)
13.                offspring_population.extend([child1, child2])
        // Update population with new individuals
14.            population = offspring_population
        // Return the best feature subset found
15.            best_chromosome = SelectBestChromosome(population, fitness)
16.            best_features = Decode Quantum Bits(best_chromosome)
17.            return best_features
    // Quantum Particle Swarm Optimization (QPSO) for Hyperparameter Tuning
18.    function QPSO_Hyperparameter_Tuning(data, features, swarm_size, iterations, hyperparameter_ranges):
        // Initialize swarm with quantum particles representing hyperparameters
19.        Initialize swarm of particles each with positions and velocities in hyperparameter_ranges
20.        for iteration in range(iterations):
21.            for particle in swarm:
                // Evaluate fitness of each particle using QSVM with current features and hyperparameters
22.                hyperparameters = DecodeParticle(particle)
23.                fitness = EvaluateFitness(particle, data, features, hyperparameters)
24.                UpdateParticleBest(particle, fitness)
                // Update global best particle position
25.                global_best_position = SelectGlobalBest(swarm)
        // Update particle positions based on quantum behaviors

```

Table 8. Pseudocode for quantum (QGA-QPSO-QSVM).

Quantum error mitigation During the training of the quantum-inspired model (QGA-QPSO-QSVM), error mitigation strategies were critical due to the inherent noise in quantum simulations. While the current implementation uses classical simulators (Qiskit Aer), the authors incorporated quantum error mitigation techniques such as measurement error correction and zero-noise extrapolation to approximate the impact of hardware noise. Measurement errors were mitigated by calibrating the readout error probabilities and applying correction matrices to the measurement outcomes. For gate errors, the authors employed randomized compiling to average out coherent errors and used dynamical decoupling sequences to suppress decoherence. These steps ensured that the quantum-inspired optimizations (e.g., QGA's rotation gates, QPSO's quantum tunnelling) remained robust despite simulated noise. Future deployments on actual quantum hardware will leverage these techniques alongside error-correcting codes (e.g., surface codes) to further enhance reliability.

The pseudocode in Table 8 outlines the integration of these quantum operations. This pseudocode provides a complete representation of the quantum model, including post-optimization steps (e.g., QSVM classification) to ensure clarity and reproducibility. These steps are essential for demonstrating how the optimized features

```

26. for particle in swarm:
27.     particle.position = QuantumUpdatePosition(particle, global_best_position)
28.     particle.velocity = QuantumUpdateVelocity(particle, global_best_position)
29.     // Return the best hyperparameters found
30.     best_particle = SelectGlobalBest(swarm)
31.     best_hyperparameters = DecodeParticle(best_particle)
32.     return best_hyperparameters
33.     // Quantum Support Vector Machine (QSVM) for Classification
34.     function QSVM_Classification(data, features, hyperparameters):
35.         // Train QSVM model using the selected features and optimized hyperparameters
36.         model = TrainQSVM(data, features, hyperparameters)
37.         return model
38.     // Main function integrating QGA, QPSO, and QSVM
39.     function QSVM_QGA_QPSO_Model(data, population_size, generations, qubit_length, swarm_size, iterations,
40.                                   hyperparameter_ranges):
41.         Step 1: Feature Selection using QGA
42.         best_features = QGA_Feature_Selection(data, population_size, generations, qubit_length)
43.         Step 2: Hyperparameter Tuning using QPSO
44.         best_hyperparameters = QPSO_Hyperparameter_Tuning(data, best_features, swarm_size, iterations, hyperparameter_ranges)
45.         Step 3: Classification using QSVM
46.         trained_model = QSVM_Classification(data, best_features, best_hyperparameters)
47.         return trained_model
48.     data = LoadDataset("Combineheartdiseasedataset.csv")
49.     population_size = 50
50.     generations = 100
51.     qubit_length = 6
52.     swarm_size = 30
53.     iterations = 50
54.     hyperparameter_ranges = DefineHyperparameterRanges()
55.     trained_model = QSVM_QGA_QPSO_Model(data, population_size, generations, qubit_length, swarm_size, iterations,
56.                                         hyperparameter_ranges)

```

Fig. 8. (continued)

and hyperparameters are utilized in practice. Here, Steps 1 to 36 cover the QGA-based feature selection and QPSO-based hyperparameter tuning, and Steps 37 onwards detail the QSVM classification phase, showing how the optimized parameters are applied to classify heart disease cases. The QGA-QPSO-QSVM framework demonstrates superior performance by harnessing quantum parallelism and entanglement, as validated in Sect. 5.

Result and discussion

The research work introduced a novel hybrid model named GA-PSO-SVM for heart disease diagnosis, leveraging advanced techniques in feature selection, hyperparameter tuning, and classification. This model integrates the GA for selecting the most relevant features, Particle Swarm Optimization (PSO) for fine-tuning the SVM parameters, and SVM for the classification task. The dataset comprises 918 samples, categorized into healthy individuals and potential heart disease patients as mentioned in the data description. For evaluation, a train-test split of 80:20 was applied, with 734 samples used for training and 184 samples reserved for testing. To enhance robustness, the model employed 10-fold cross-validation, ensuring that each data point is used both in training and testing, thus mitigating overfitting and providing a reliable performance estimate. The study distinguishes itself by applying both classical (CGA-CPSO-CSVM) and quantum-inspired (QGA-QPSO-QSVM) versions of these algorithms. The results of the classical, and the quantum versions are discussed in the following experiments 1 and 2 respectively.

Experiment 1: classical model (CGA-CPSO-CSVM)

When the classical version (CGA-CPSO-CSVM) was applied to the combined heart disease dataset, using 184 test samples (20% of the total data), 9 out of 11 features were selected by CGA. Further hyperparameter tuning by CPSO and classification by CSVM were conducted. The confusion matrix and performance metrics have been

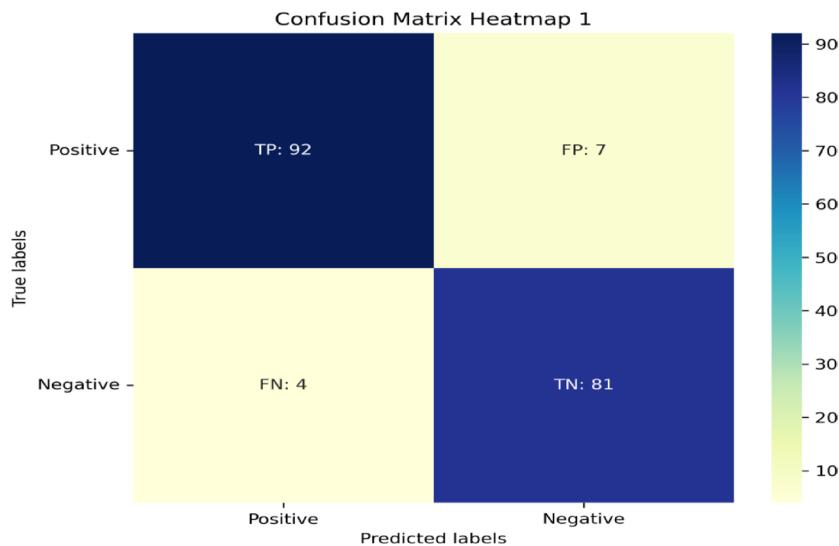


Fig. 20. Confusion matrix of the classical model.

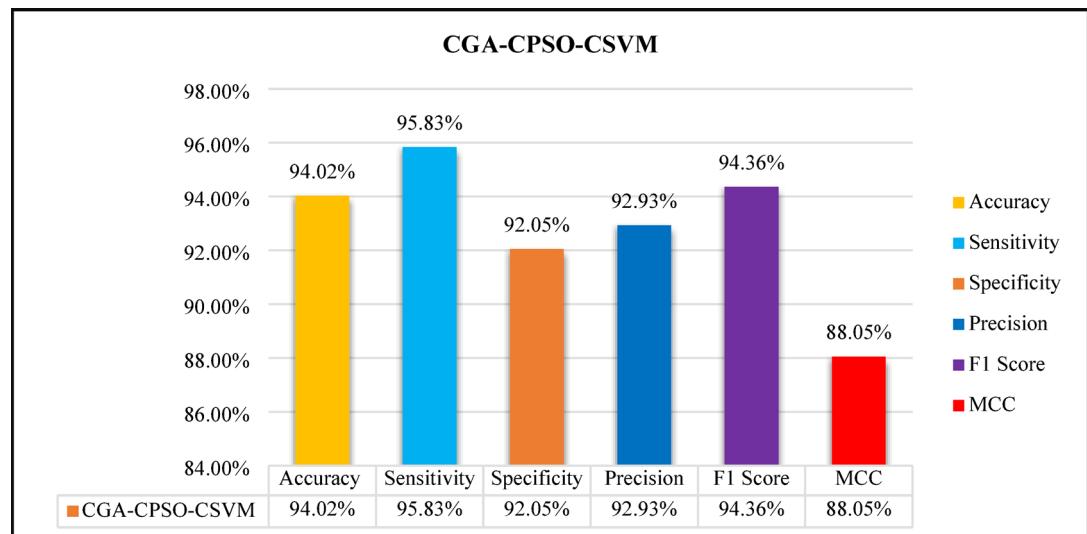


Fig. 21. Performance metrics of the classical model.

obtained, as shown in Figs. 20 and 21 respectively. The confusion matrix for the classical model (CGA-CPSO-CSVM) indicates strong performance metrics, with 92 true positives, 81 true negatives, 7 false positives, and 4 false negatives as shown in Fig. 20. This results in an accuracy of 94.02%, a precision of 92.93%, a sensitivity of 95.83%, an F1-score of 94.36%, a specificity of 92.05%, and an MCC of 88.05% as shown in Fig. 21. These high values across all metrics reflect the classical model's effectiveness in correctly identifying positive and negative cases, demonstrating its robustness and reliability in classification tasks.

Figures 22 and 23 depicts the ROC curves for the classical models and the ROC curve with tenfold cross-validation, respectively. The train AUC (Area Under the Curve) and test AUC of the classical model are 95% and 93%, respectively, as shown in Fig. 22. The tenfold cross-validation ROC curves for each fold are as follows: Fold1 (AUC = 95%), Fold2 (AUC = 93%), Fold3 (AUC = 94%), Fold4 (AUC = 94%), Fold5 (AUC = 94%), Fold6 (AUC = 94%), Fold7 (AUC = 94%), Fold8 (AUC = 94%), Fold9 (AUC = 94%), and Fold10 (AUC = 94%), with an average AUC of 94%, as shown in Fig. 23. These ROC curves further illustrate the model's ability to discriminate between positive and negative classes effectively, highlighting its robustness and reliability in classification tasks through both individual and cross-validated assessments.

Experiment 2: quantum model (QGA-QPSO-QSVM)

The quantum model (QGA-QPSO-QSVM) was executed on classical simulators (Qiskit Aer) with quantum-inspired circuits. Trained on 734 samples and tested on 184 samples (80:20 split), the quantum version (QGA-QPSO-QSVM) was applied to the combined heart disease dataset, 7 out of 11 features were selected by the QGA.

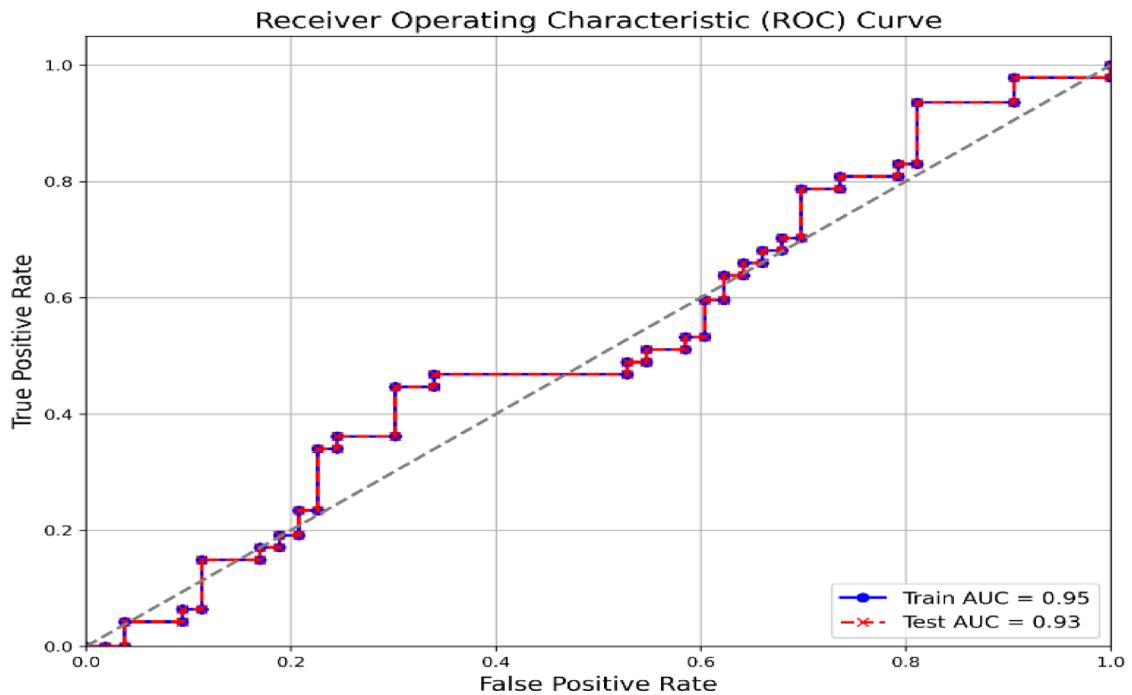


Fig. 22. ROC curve of the classical model.

QGA reduced the feature set by 37%, while QPSO optimized QSVM hyperparameters. Further hyperparameter tuning was conducted by the QPSO algorithm, and classification was performed using the QSVM with 7 qubits. The results, including the confusion matrix and performance metrics, are presented in Figs. 24 and 25, respectively. The confusion matrix for the quantum model revealed impressive performance metrics, with 95 true positives, 85 true negatives, 3 false positives, and 1 false negative. This results in an accuracy of 97.83%, precision of 96.94%, sensitivity of 98.96%, F1-score of 97.94%, specificity of 96.59%, and MCC of 95.66. These metrics highlighted the quantum model's strong ability to accurately discriminate between positive and negative cases, demonstrating its robustness and reliability in classification tasks, as further supported by the detailed ROC analysis presented in Figs. 26 and 27.

Figures 26 and 27 display the ROC curves for the quantum models and their respective tenfold cross-validation. In Fig. 26, the classical model achieved a train AUC of 98% and a test AUC of 97%. The tenfold cross-validation results in Fig. 27 shows AUC values for each fold: The tenfold cross-validation ROC curves for each fold are as follows: Fold1 (AUC=98%), Fold2 (AUC=97%), Fold3 (AUC=99%), Fold4 (AUC=98%), Fold5 (AUC=97%), Fold6 (AUC=98%), Fold7 (AUC=97%), Fold8 (AUC=97%), Fold9 (AUC=97%), and Fold10 (AUC=96%), with an average AUC of 97%. These ROC curves demonstrated the quantum models' strong performance in distinguishing between positive and negative classes, highlighting their reliability and robustness in classification tasks across both individual evaluations and cross-validation.

The superior performance of the quantum model (97.83% accuracy with only 7 features) demonstrates significant potential for clinical implementation. By focusing on key interpretable features like ST segment depression (Oldpeak) and exercise-induced angina - parameters already familiar to cardiologists - the model achieves high diagnostic accuracy while maintaining clinical relevance. This streamlined feature set offers three practical advantages: (1) Reduced computational complexity enables real-time analysis in emergency settings, (2) Alignment with existing diagnostic protocols (e.g., ACC/AHA guidelines) enhances physician trust and adoption, and (3) Simplified input requirements facilitate integration with existing EHR systems and wearable monitoring devices. Notably, while the classical model retained 9 features with slightly lower accuracy (94.02%), the quantum approach demonstrates that carefully selected, clinically meaningful features can simultaneously improve both accuracy and interpretability - a critical combination for medical decision support systems.

Diagnostic metrics for classical and quantum models

Furthermore, for the additional analysis the diagnostic performance metrics LR+ (Positive Likelihood Ratio), LR- (Negative Likelihood Ratio), and DOR (Diagnostic Odds Ratio) for classical (CGA-CPSO-CSVM) and quantum (QGA-QPSO-QSVM) models as shown in Fig. 28, QGA-QPSO-QSVM consistently outperformed CGA-CPSO-CSVM across all measures. QGA-QPSO-QSVM exhibited a higher LR+ (29.03 vs. 12.05), indicating a greater ability to correctly identify positive cases, and a lower LR- (0.01 vs. 0.05), indicating a better ability to rule out negative cases. Moreover, QGA-QPSO-QSVM had a substantially higher DOR (2691.67 vs. 266.14), highlighting its superior overall diagnostic accuracy and reliability compared to the classical model. These findings suggest that the quantum model is more effective in both detecting the presence of a condition and accurately excluding its absence, underscoring its advanced diagnostic performance capabilities.

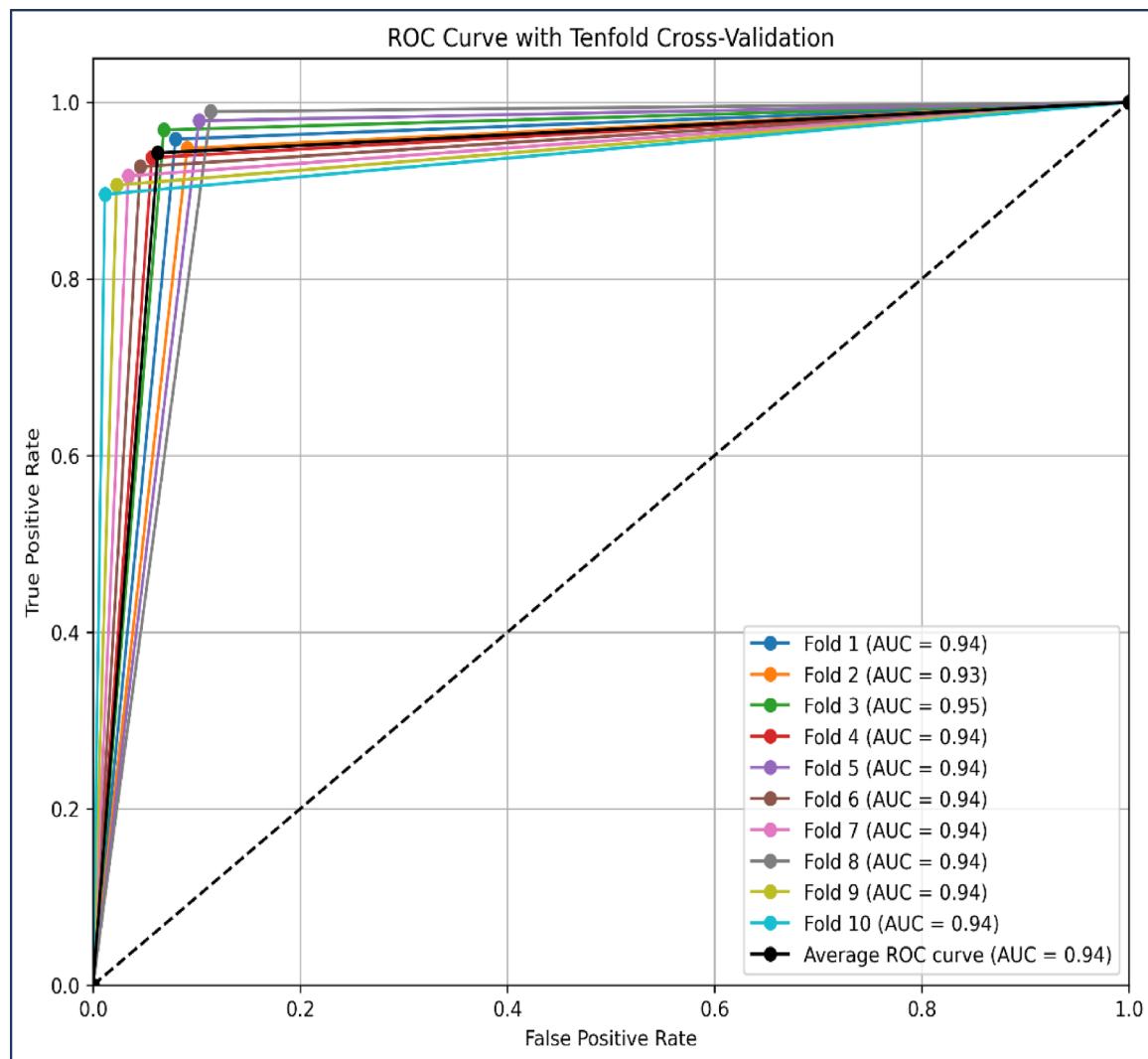


Fig. 23. ROC curve with ten-fold cross-validation of the classical model.

The experimental results from both trials decisively establish the quantum model's superiority over the classical model across a comprehensive set of performance metrics, encompassing Accuracy, F1-score, Precision, Sensitivity, Specificity, and ROC with 10-fold cross-validation. Quantitative evaluation of LR+, LR-, and DOR consistently demonstrates superior performance metrics for the quantum model (QGA-QPSO-QSVM) compared to the classical model (CGA-CPSO-CSVM). These findings robustly validate the quantum model's enhanced capability in accurately identifying positive cases and correctly excluding negative cases, thereby underscoring its superior diagnostic accuracy and reliability over its classical counterpart. For further validation, both the quantum (QGA-QPSO-QSVM) and classical (CGA-CPSO-CSVM) models were individually applied to the Cleveland (303 observations)¹⁹ heart disease dataset. Consistently, the same trend of results was observed, reaffirming the quantum model's superiority over the classical model across all evaluated metrics.

Evaluation of feature selection effectiveness

In this subsection, the authors evaluated the performance of the feature selection (FS) process using CGA and QGA. The FS process enhanced the performance of the proposed approach compared to using the entire feature set. Specifically, the feature sets of the combined dataset were reduced by 19% using CGA and 37% using QGA. From Table 9, it is evident that the proposed approach significantly decreased the number of features. The features selected by CGA amounted to 9 features, whereas QGA selected 7 features out of 11. Table 9 summarizes the overall measurement results and shows that the features selected by QGA produced accuracy increments of 5–10% compared to existing studies and the features selected by CGA.

Comparative analysis with the existing state-of-arts

The comparison Table 10 summarizes several studies that utilized different feature selection methods and hyperparameter tuning techniques, alongside various classification algorithms, highlighting their respective accuracies. Key observations included the prevalence of GA for feature selection and PSO for hyperparameter

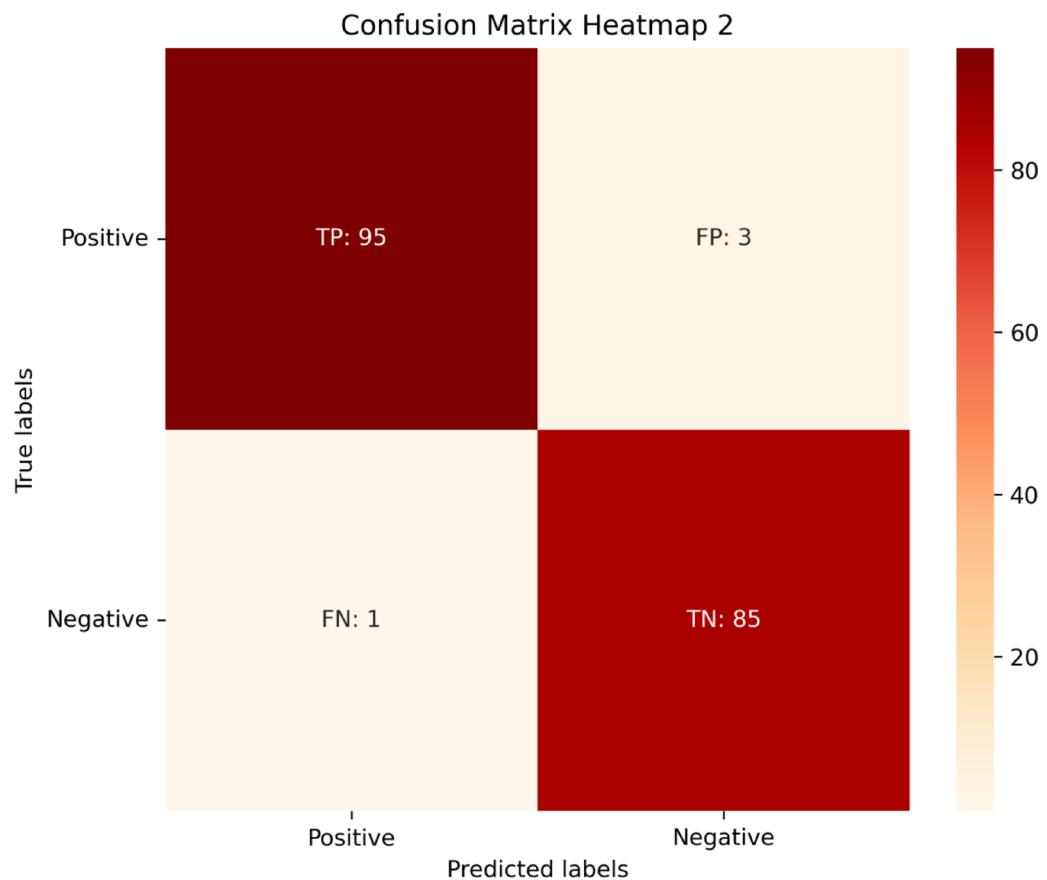


Fig. 24. Confusion matrix of the quantum model.

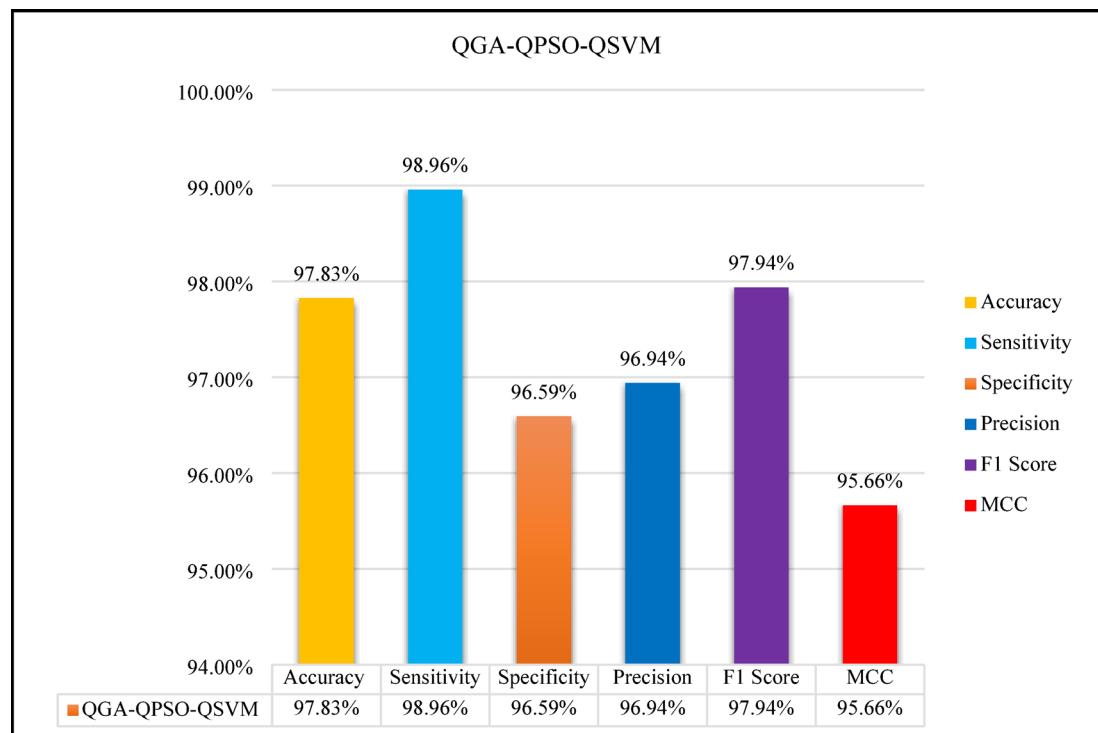


Fig. 25. Performance metrics of the quantum model.

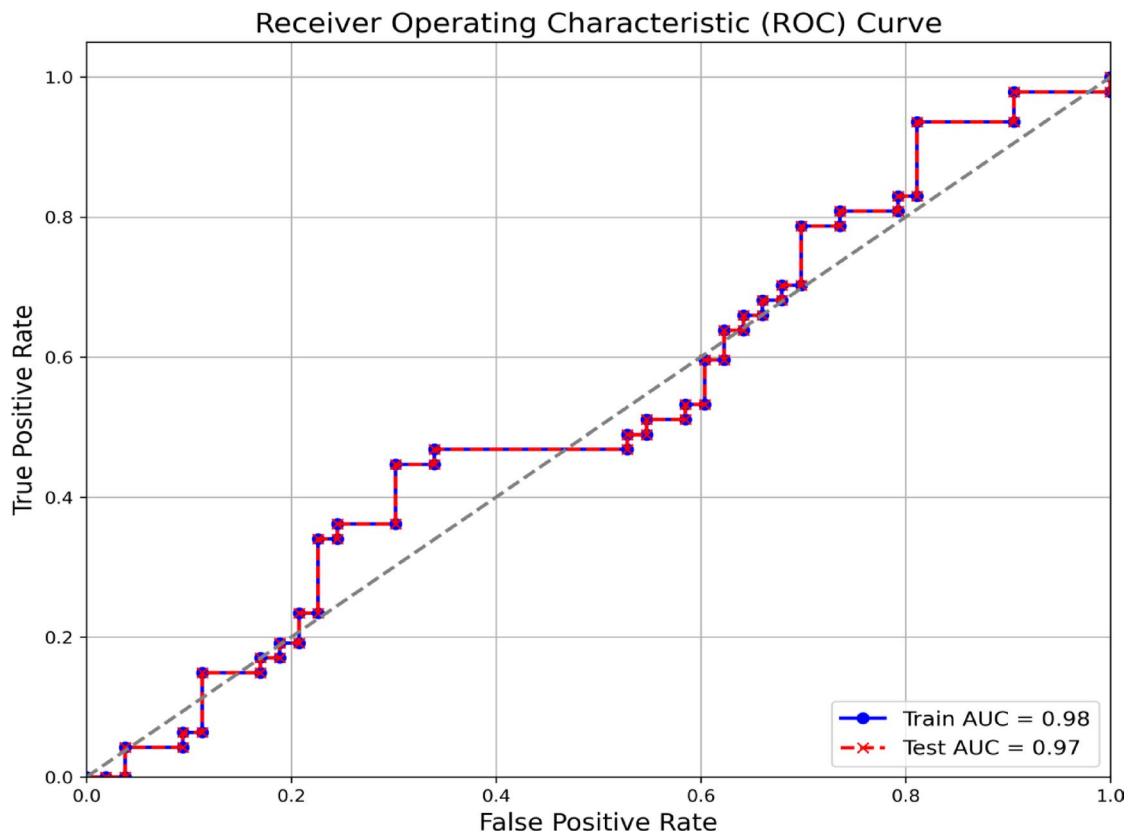


Fig. 26. ROC curve of the quantum model.

tuning in many studies, often paired with SVM or Random Forest (RF) classifiers. Noteworthy results included accuracies ranging from 81.42 to 95.6% across different methodologies. The proposed classical model introduces CGA for feature selection, CPSO for hyperparameter tuning, and CSVM for classification, achieved a competitive accuracy of 94.02%. In contrast, the proposed quantum model incorporated QGA and QPSO for feature selection and hyperparameter tuning, respectively, coupled with QSVM for classification, showcased a superior accuracy of 97.83%. These findings highlighted the efficacy of advanced optimization techniques and the potential of quantum computing in improving predictive performance in machine learning tasks, emphasizing their role in advancing state-of-the-art methodologies.

Time complexity

In this subsection, the comparison of the time complexity of the proposed approaches, CGA-PSO-SVM and QGA-QPSO-QSVM, with various GA-based models is presented. Table 10 presents the comparison of computational time among these approaches on the combined heart disease dataset. This table details the complexity time (including FS and classification), and the prediction accuracy. The quantum model (QGA-QPSO-QSVM) was executed on classical simulators (Qiskit Aer) with quantum-inspired circuits, achieving a runtime of 413.57 s, compared to the classical model's 362.92 s. Although the computational time for the proposed approaches, CGA-PSO-SVM and QGA-QPSO-QSVM, is quite high, Table 10 clearly shows that the proposed methods achieved the highest prediction accuracy compared to other GA-based methods. While the quantum model exhibits higher computational overhead due to quantum superposition and entanglement simulations, its benefits justify the trade-off.

Superior accuracy (97.83% vs. 94.02%) The quantum model's ability to explore high-dimensional feature spaces via quantum parallelism enhances predictive performance.

Feature reduction (37% vs. 19%) QGA's quantum-inspired optimization efficiently identifies clinically relevant features, reducing redundancy.

Diagnostic robustness Quantum kernels in QSVM capture non-linear relationships better than classical SVMs, reflected in higher LR+ (29.03 vs. 12.05) and DOR (2691.67 vs. 266.14).

Future hardware advancements (e.g., IBM QPU deployments) are expected to mitigate runtime limitations while preserving quantum advantages.

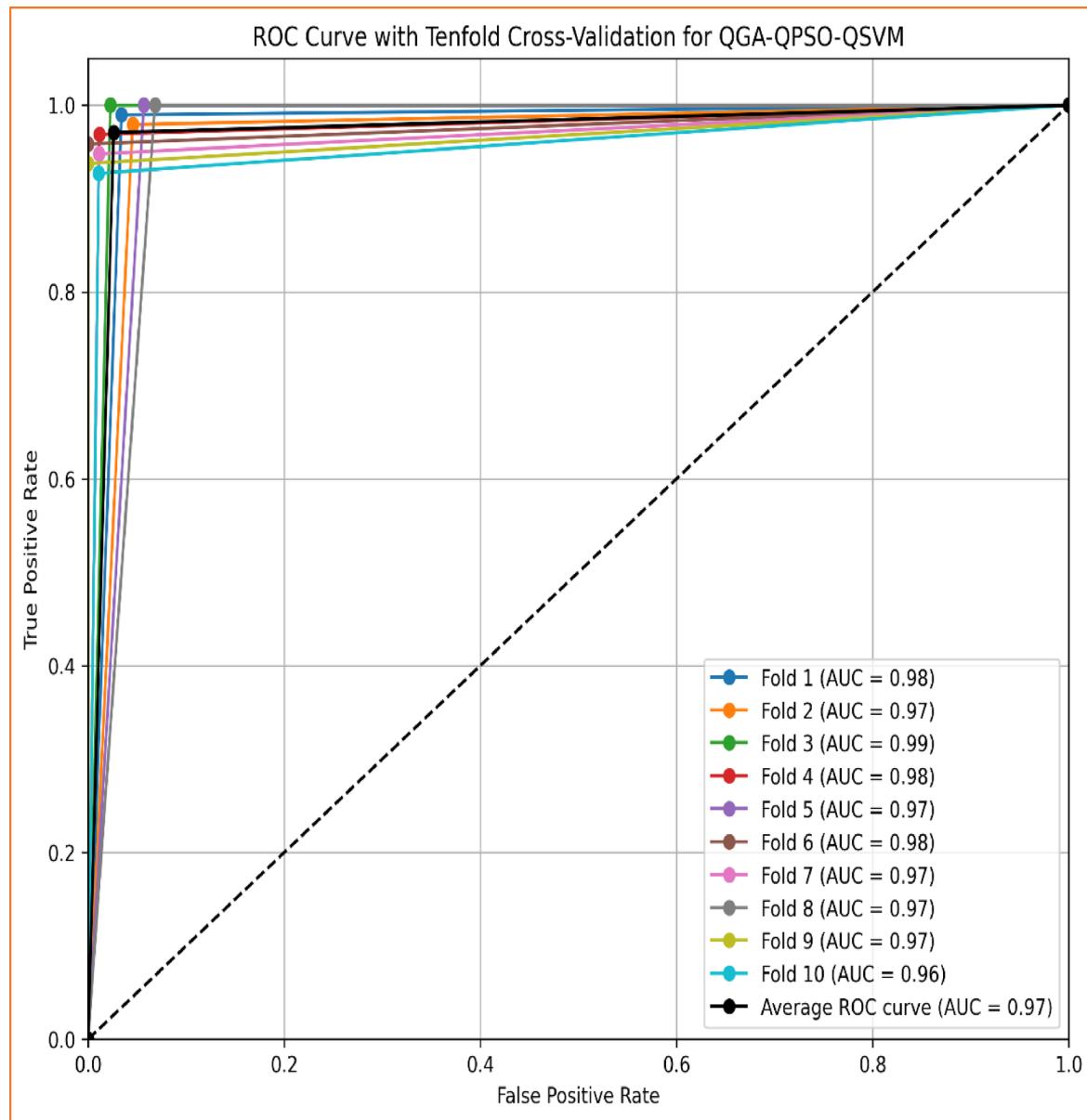


Fig. 27. ROC curve with ten-fold cross-validation of the quantum model.

Conclusion and future scope

This study successfully demonstrated the efficacy of a hybrid approach that integrated classical and quantum-inspired machine learning techniques for enhancing heart disease prediction. The proposed framework leverages a combined dataset, employed feature selection using CGA and QGA to reduce dimensionality, and utilized CPSO and QPSO for hyperparameter optimization. The results demonstrate that the quantum-inspired model (QGA-QPSO-QSVM) outperforms the classical model (CGA-CPSO-CSVM) in terms of accuracy (97.83%, 94.02%), sensitivity (98.96%, 94.02%), specificity (96.59%, 94.02%), precision (96.94%, 94.02%), F1-score (97.94%, 94.02%), MCC (95.66%, 94.02%), LR+ (29.027, 12.047%), LR- (0.0107, 0.0452), and DOR (2691.666, 266.1428), respectively. Notably, the QGA-QPSO-QSVM model achieved an accuracy of 97.83%, surpassing the performance of existing state-of-the-art methods^{17,23,28–35}. The feature selection process using QGA significantly reduced the feature set by 37% while maintaining high accuracy, indicating its effectiveness in identifying relevant predictors. While the quantum-inspired model exhibits superior performance, it also comes with an increased computational cost compared to the classical model. The computational overhead of quantum-inspired algorithms arises from their reliance on quantum superposition and entanglement, which, while powerful, demand more resources in terms of both time and hardware. For instance, the proposed QGA-QPSO-QSVM model required 413.57 s for execution, whereas the classical CGA-CPSO-CSVM model took 362.92 s (as shown in Table 10). This trade-off between accuracy and computational efficiency must be carefully considered in practical applications. In scenarios where real-time or resource-constrained predictions are

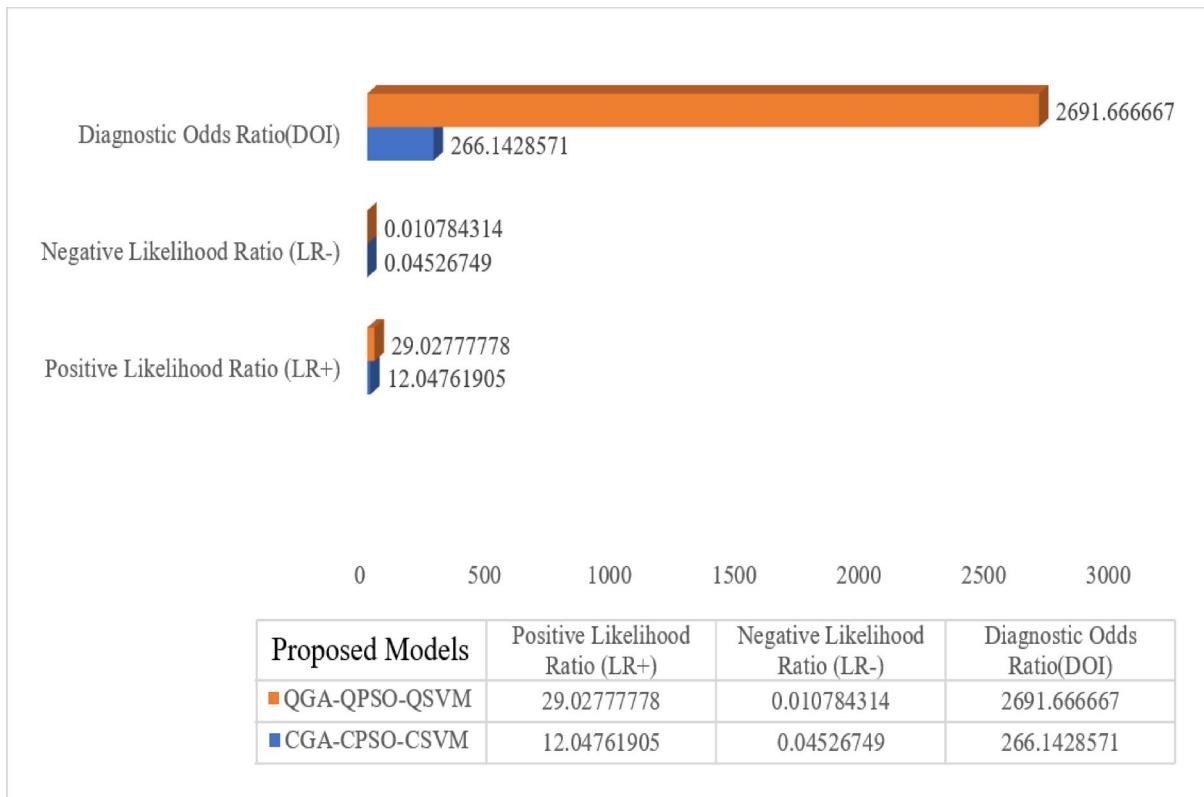


Fig. 28. Diagnostic metrics of both classical and quantum models.

Model	Total Feature	Selected Feature	Reduced (%) = $\left(1 - \frac{\text{Selected Features}}{\text{Total Features}}\right) \times 100$
CGA	11	9	19
QGA	11	7	37

Table 9. Feature selection details of classical and quantum models.

Authors	Feature Selection Methods	Hyperparameter Tuning	Classification Methods	Accuracy (%)	Time (s)	Sensitivity (%)	Precision (%)	F1-Score (%)
Vijayshree et al. ²⁸	Chi-squared	PSO	SVM	88.22	148	---	---	---
Shafiey et al. ²⁹	GA	PSO	RF	95.6	2525.59	94.1	96.2	95.1
Huang et al. ²³	GA	---	SVM	85.6	---	---	---	---
Tao et al. ³⁰	GA	PSO	SVM	89.88	---	---	---	---
Khammassi et al. ³¹	GA-LR	---	DT	81.42	---	79.3	82.0	80.6
Temitayo et al. ³²	GA	---	SVM	93.5	119.56	92.8	94.1	93.4
Chen et al. ³³	GA	---	SVM	95.56	---	---	---	---
Nandipati et al. ³⁴	Pearson Correlation	---	SVM / RF	82.27 / 85	117.57 / 187	80.2 / 84.1	83.4 / 85.9	81.7 / 85
Sreejith et al. ¹⁷	Wrapper Method	Ant Colony	SVM / RF	84.5 / 89.81	---	83.2 / 88.9	85.0 / 90.5	84.1 / 89.7
Abdulsalam et al. ³⁵	---	---	QSVM / QNN	88 / 86	317 / 455.51	87.5 / 85.1	88.2 / 86.3	87.8 / 85.7
Proposed Classical Model	CGA	CPSO	CSVM	94.02	362.92	93.1	94.4	93.7
Proposed Quantum Model	QGA	QPSO	QSVM	97.83	413.57	97.2	98.1	97.6

Table 10. Comparative analysis of the proposed models with the existing state of arts.

needed, classical models may still be preferable. However, for high-stakes medical diagnostics where accuracy is paramount, the quantum-inspired approach justifies its computational expense.

Future research directions include exploring the impact of different quantum-inspired optimization algorithms, investigating the use of deep learning architectures for feature extraction, and developing explainable

AI models to enhance interpretability and clinical trust. Additionally, integrating real-time data streams and incorporating patient-specific factors could further improve the accuracy and personalized nature of heart disease prediction. Finally, the application of this framework can be extended to other medical domains, such as predicting the risk of other cardiovascular diseases, cancer, and neurological disorders. By leveraging the power of quantum-inspired algorithms and incorporating real-time data streams, this research can pave the way for personalized and proactive healthcare solutions, ultimately improving patient outcomes and reducing the burden of chronic diseases.

Data availability

In this research study an open-source heart disease datasets have been used, which is freely available on the UCI Machine Learning repository (<https://archive.ics.uci.edu/dataset/45/heart+disease>).

Received: 16 February 2025; Accepted: 1 July 2025

Published online: 11 July 2025

References

1. Mohan, S., Thirumalai, C. & Srivastava, G. Effective heart disease prediction using hybrid machine learning techniques. *IEEE Access*. **7**, 81542–81554. <https://doi.org/10.1109/ACCESS.2019.2923707> (2019).
2. Long, N. C., Meesad, P. & Unger, H. A highly accurate firefly based algorithm for heart disease prediction. *Expert Syst. Appl.* **42**, 8221–8231. <https://doi.org/10.1016/j.eswa.2015.06.024> (2015).
3. Dwivedi, A. K. Performance evaluation of different machine learning techniques for prediction of heart disease. *Neural Comput. Appl.* **29** (10), 685–693. <https://doi.org/10.1007/s00521-016-2604-1> (2018).
4. Reddy, K. V. V. et al. Prediction of Heart Disease Risk Using Machine Learning with Correlation-based Feature Selection and Optimization Techniques, in *7th International Conference on Signal Processing and Communication, ICSC 2021*, Institute of Electrical and Electronics Engineers Inc., 2021, pp. 228–233., Institute of Electrical and Electronics Engineers Inc., 2021, pp. 228–233. (2021). <https://doi.org/10.1109/ICSC53193.2021.9673490>
5. Heidari, H. & Hellstern, G. Early heart disease prediction using hybrid quantum classification, in IEEE International Conference on Quantum Computing and Engineering (QCE), Bellevue, WA, USA, 2023, pp. 206–212, (2023). <https://doi.org/10.1109/QCE57785.2023.00042>
6. Ogundepo, E. A. & Yahya, W. B. Performance analysis of supervised classification models on heart disease prediction. *Innov. Syst. Softw. Eng.* **19** (1), 129–144. <https://doi.org/10.1007/s11334-022-00524-9> (2023).
7. Paul, B. & Karn, B. Heart disease prediction using scaled conjugate gradient backpropagation of artificial neural network. *Soft Comput.* **27** (10), 6687–6702. <https://doi.org/10.1007/s00500-022-07649-w> (2023).
8. Ayon, S. I., Islam, M. M. & Hossain, M. R. Coronary artery heart disease prediction: A comparative study of computational intelligence techniques. *IETE J. Res.* **68** (4), 2488–2507. <https://doi.org/10.1080/03772063.2020.1713916> (2022).
9. GÜLLÜ, M., AKCAYOL, M. A. & BARİŞÇİ, N. Machine Learning-Based Comparative Study For Heart Disease Prediction, *Advances in Artificial Intelligence Research*, vol. 2, no. 2, pp. 51–58, Sep. (2022). <https://doi.org/10.54569/air.1145616>
10. Sarra, R. R., Dinar, A. M., Mohammed, M. A. & Abdulkareem, K. H. Enhanced heart disease prediction based on machine learning and χ^2 statistical optimal feature selection model. *Designs (Basel)*. **6** (5). <https://doi.org/10.3390/designs6050087> (2022).
11. Nijaguna, G. S., Babu, J. A., Parameshchari, B. D., de Prado, R. P. & Frnda, J. Quantum fruit fly algorithm and ResNet50-VGG16 for medical diagnosis. *Appl. Soft Comput.* **136** <https://doi.org/10.1016/j.asoc.2023.110055> (2023).
12. Amraoui, K. E. et al. Machine learning algorithm for Avocado image segmentation based on quantum enhancement and Random forest, in, *2nd International Conference on Innovative Research in Applied Science, Engineering and Technology, IRASET 2022*, Institute of Electrical and Electronics Engineers Inc., 2022, (2022). <https://doi.org/10.1109/IRASET52964.2022.9738360>
13. Sridevi, S., Kanimozhi, T., Bhattacharjee, S., Shahwar, D. & Sekhar Reddy, K. S. Quantum Transfer Learning for Diagnosis of Diabetic Retinopathy, in *International Conference on Innovative Trends in Information Technology, ICITIIT 2022*, Institute of Electrical and Electronics Engineers Inc., 2022., Institute of Electrical and Electronics Engineers Inc., 2022. (2022). <https://doi.org/10.1109/ICITIIT54346.2022.9744184>
14. Shi, Y., Wong, W. K., Goldin, J. G., Brown, M. S. & Kim, G. H. J. Prediction of progression in idiopathic pulmonary fibrosis using CT scans at baseline: A quantum particle swarm optimization - Random forest approach. *Artif. Intell. Med.* **100** <https://doi.org/10.1016/j.artmed.2019.101709> (2019).
15. Wang, Y., Wang, Y., Chen, C., Jiang, R. & Huang, W. Development of variational quantum deep neural networks for image recognition. *Neurocomputing* **501**, 566–582. <https://doi.org/10.1016/j.neucom.2022.06.010> (2022).
16. Dutta, P., Paul, S. & Majumder, M. An efficient SMOTE based machine learning classification for prediction & detection of PCOS, (2021). <https://doi.org/10.21203/rs.3.rs-1043852/v1>
17. Sreejith, S., Khanna Nehemiah, H. & Kannan, A. A clinical decision support system for polycystic ovarian syndrome using red deer algorithm and random forest classifier. *Healthc. Analytics.* **2** <https://doi.org/10.1016/j.health.2022.100102> (2022).
18. Nasim, S., Almutairi, M. S., Munir, K., Raza, A. & Younas, F. A novel approach for polycystic ovary syndrome prediction using machine learning in bioinformatics. *IEEE Access*. **10**, 97610–97624. <https://doi.org/10.1109/ACCESS.2022.3205587> (2022).
19. Detrano, W. S. M. P. R., Janosi, A., Steinbrunn, M. A. S. E. & Guppy, M. *Heart Disease Data Set*, UCI Machine Learning Repository, [Online]. (1988). Available: <https://archive.ics.uci.edu/ml/datasets/Heart+Disease>
20. Michie, D., Spiegelhalter, D. J., Taylor, C. C. & Campbell, J. *Statlog (Heart) Data Set*, UCI Machine Learning Repository, [Online]. (1992). Available: [https://archive.ics.uci.edu/ml/datasets/statlog+\(heart\)](https://archive.ics.uci.edu/ml/datasets/statlog+(heart))
21. Shankar, K., Lakshmanaprabhu, S. K., Gupta, D., Maselino, A. & de Albuquerque, V. H. C. Optimal feature-based multi-kernel SVM approach for thyroid disease classification, 2020, *Springer*. <https://doi.org/10.1007/s11227-018-2469-4>
22. Mathew, T. V. & National Programme on Technology Enhanced Learning (NPTEL). *Genetic Algorithm*,, Indian Institute of Technology (IIT) Bombay, 2003, [Online]. Available: <https://nptel.ac.in/courses/105101010/downloads/lecnotes/mod12.pdf>
23. Huang, C. L. & Wang, C. J. A GA-based feature selection and parameters optimization for support vector machines. *Expert Syst. Appl.* **31** (2), 231–240. <https://doi.org/10.1016/j.eswa.2005.09.024> (2006).
24. Wang, D., Tan, D. & Liu, L. Particle swarm optimization algorithm: an overview. *Soft Comput.* **22** (2), 387–408. <https://doi.org/10.1007/s00500-016-2474-6> (2018).
25. Havlíček, V. et al. Supervised learning with quantum-enhanced feature spaces. *Nature* **567** (7747), 209–212. <https://doi.org/10.1038/s41586-019-0980-2> (2019).
26. Rainio, O., Teuho, J. & Klén, R. Evaluation metrics and statistical tests for machine learning. *Sci. Rep.* **14** (1). <https://doi.org/10.1038/s41598-024-56706-x> (2024).
27. Wong, T. T. & Yeh, P. Y. Reliable Accuracy Estimates from k-Fold Cross Validation, *IEEE Trans Knowl Data Eng*, vol. 32, no. 8, pp. 1586–1594, (2020). <https://doi.org/10.1109/TKDE.2019.2912815>

28. Vijayashree, J. & Sultana, H. P. A machine learning framework for feature selection in heart disease classification using improved particle swarm optimization with support vector machine classifier. *Program. Comput. Softw.* **44** (6), 388–397. <https://doi.org/10.1134/S0361768818060129> (2018).
29. El-Shafiey, M. G., Hagag, A., El-Dahshan, E. S. A. & Ismail, M. A. A hybrid GA and PSO optimized approach for heart-disease prediction based on random forest. *Multimed Tools Appl.* **81** (13), 18155–18179. <https://doi.org/10.1007/s11042-022-12425-x> (2022).
30. Tao, Z., Huiling, L., Wenwen, W. & Xia, Y. GA-SVM based feature selection and parameter optimization in hospitalization expense modeling. *Appl. Soft Comput. J.* **75**, 323–332. <https://doi.org/10.1016/j.asoc.2018.11.001> (2019).
31. Khammassi, C. & Krichen, S. A GA-LR wrapper approach for feature selection in network intrusion detection. *Comput. Secur.* **70**, 255–277. <https://doi.org/10.1016/j.cose.2017.06.005> (2017).
32. Temitayo, F., Stephen, O. & Abimbola, A. Hybrid GA-SVM for efficient feature selection in e-mail classification. *Comput. Eng. Intell. Syst.* **3** (3), 17–28 (2012).
33. Chen, L. H. & Hsiao, H. D. Feature selection to diagnose a business crisis by using a real GA-based support vector machine: an empirical study. *Expert Syst. Appl.* **35** (3), 1145–1155. <https://doi.org/10.1016/j.eswa.2007.08.010> (2008).
34. Nandipati, S. C. & Ying, C. X. Polycystic ovarian syndrome (PCOS) classification and feature selection by machine learning techniques. *Appl. Math. Comput. Intell. (AMCI)* **9**, 65–74 (2020).
35. Abdulsalam, G., Meshoul, S. & Shaiba, H. Explainable heart disease prediction using Ensemble-Quantum machine learning approach. *Intell. Autom. Soft Comput.* **36** (1), 761–779. <https://doi.org/10.32604/iasc.2023.032262> (2023).

Author contributions

Ankur Kumar: Conceptualization, methodology, data collection, formal analysis, and manuscript drafting. Sanjay Dhanka: Methodology, formal analysis, investigation, supervision, validation, review, and editing of the manuscript. Abhinav Sharma: Data preprocessing, statistical analysis. Rohit Bansal: Funding acquisition and project administration. Mohammad Fahlevi (Corresponding Author): Funding acquisition, project administration, and manuscript enhancement. Fazla Rabby: Funding acquisition and project administration. Mohammed Al-juaid: Funding acquisition and project administration.

Funding

This work has been supported by King Saud University, Riyadh, Saudi Arabia, through the Researchers Supporting Project number (ORF-2025-481).

Declarations

Competing interests

The authors declare no competing interests.

Additional information

Correspondence and requests for materials should be addressed to M.F. or M.A.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Open Access This article is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License, which permits any non-commercial use, sharing, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if you modified the licensed material. You do not have permission under this licence to share adapted material derived from this article or parts of it. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by-nc-nd/4.0/>.

© The Author(s) 2025