The Journal of Supercomputing (2025) 81:827
https://doi.org/10.1007/s11227-025-07266-y

# QEKLR: quantum-enhanced kernel logistic regression for classification

**Shreshtha Misra**[1] · **Poonam Rani**[1]

Accepted: 30 March 2025


## Abstract

Classification is fundamental in machine learning (ML) applications such as medical diagnosis, financial analysis, and image recognition. Quantum machine learning shows promise in enhancing feature representation and computational efficiency. However, challenges like the noisy intermediate-scale quantum hardware limitations, lack of interpretability, and barren plateau issue in parameterized quantum circuits hinder its practical adoption. To address these issues, this work proposes a novel hybrid approach called quantum-enhanced kernel logistic regression (QEKLR) for classification tasks, which combines quantum computation with classical logistic regression (LR) model for improved classification performance. In our QEKLR model, classical data (CD) are first transformed to quantum states (QS) using shallow-depth ZZFeatureMap, and then, fidelity kernel is employed to compute the dot product between feature vectors, resulting in a trained quantum kernel. Subsequently to predict new instances, the quantum kernel is integrated into a classical LR model. We evaluated the proposed model's effectiveness using four datasets: synthetic, Iris, Statlog Heart Disease (HD), and Ecoli datasets achieving accuracy rates of 100%, 100%, 94.87%, and 71%, respectively. In addition, experimental results reveal that the proposed QEKLR model surpasses classical ML models, achieving the MCC 0.88 and AUC-ROC 0.98 for the Statlog HD dataset demonstrating strong classification capability. However, on the Ecoli dataset, the model's performance was comparable to existing ML models, indicating balanced performance across datasets while maintaining robust classification ability and effective class distinctions.

**Keywords** Quantum computing · QML · Superposition · Entanglement · QEKLR

✉ Shreshtha Misra
shreshtha.misra.phd21@nsut.ac.in

Poonam Rani
poonam.rani@nsut.ac.in

[1] Department of Computer Science and Engineering, Netaji Subhas University of Technology, Azad Hind Fauj Marg, Dwarka, New Delhi 110078, India

Published online: 04 May 2025

**Abbreviations**

| | |
|---|---|
| $Acc_{model}$ | Model accuracy |
| CART | Classification and regression trees |
| CNOT Gate | Controlled not gate |
| FM | Feature map |
| GA | Genetic algorithm |
| H Gate | Hadamard gate |
| HD | Heart Disease |
| HHL | Harrow–Hassidim–Lloyd |
| KLR | Kernel logistic regression |
| IQR | Interquartile range |
| LR | Logistic regression |
| $CM_{model}$ | Confusion matrix of the model |
| $MCC_{model}$ | Matthews correlation coefficient of the model |
| ML | Machine learning |
| NISQ | Noisy intermediate-scale quantum |
| $Prec_{model}$ | Model precision |
| PSO | Particle swarm optimization |
| PQC | Parameterized quantum circuit |
| QBM | Quantum Boltzmann machine |
| QDT | Quantum decision tree |
| QC | Quantum computing |
| QCNN | Quantum convolutional neural network |
| QEKLR | Quantum-enhanced kernel logistic regression |
| QFT | Quantum Fourier transform |
| QGAN | Quantum generative adversarial networks |
| QKNN | Quantum K-nearest neighbor |
| QML | Quantum machine learning |
| QNN | Quantum neural network |
| QS | Quantum states |
| QSVC | Quantum support vector classifier |
| RF | Random forest |
| RBF | Radial basis function |
| SVM | Support vector machine |
| VQC | Variational quantum classifier |
| VQASVM | Variational quantum approximate support vector machine |
| $F1_{model}$ | F1 score of the model |
| $Sen_{model}$ | Model sensitivity |
| $Spec_{model}$ | Model specificity |

## 1 Introduction

Classification problems are among the most widely studied problems in the field of machine learning (ML), and they are applied in various practical areas such as disease diagnosis, financial forecasting, image recognition, and spam filtering. These

tasks involve categorizing data into pre-existing categories, which is crucial for decision-making processes across different fields [1].

A review of the conventional ML approaches [2–4] reveals significant advancements in classification accuracy. However, these methods often encounter limitations when dealing with complex and/or high-dimensional data, leading researchers to explore alternative computational models [5, 6]. QML can be described as an emerging field of study that applies the essence of quantum mechanical principles such as entanglement and superposition, to improve the efficacy of typical ML algorithms [7].

Utilizing these quantum computing concepts QML aims to solve the problems that are difficult to be solved by the classical computers, with increased performance and exponential speed-up [5, 8]. Currently, we are in the NISQ [9] era, i.e., quantum processors with fewer qubits that are insufficient to achieve quantum advantage. The NISQ era has brought up new possibilities and challenges for utilizing quantum algorithms. Several quantum algorithms were devised to show that quantum algorithms perform exponentially faster than the classical algorithms. Examples include Shor's algorithm for factorization of integers [10], one of the first types of quantum algorithms owns exponential advantage, while Grover's algorithm [11] that operates in unsorted database search has quadratic advantage. The other revolutionary applications of quantum computing include QFT [7] and the HHL algorithm, which efficiently addresses solving linear systems of equations [12].

Several quantum classification methods have been introduced in the area of QML to demonstrate the potential of QC in the field of ML. QSVC [13–15] is an extension of the traditional SVM that provides better results for a high-dimensional and complex data utilizing the properties of quantum computing. VQC [16] and the k-nearest neighbors' quantum variant QkNN [17, 18] algorithm have shown noticeable improvements in terms of time and performance. Further, quantum counterparts of the traditional neural networks and decision trees are introduced, like QNN, QDT [19, 20], and QRF [21] which incorporate the features of QC to increase the performance of the conventional models. Quantum-enhanced kernel methods [22] and quantum algorithm for logistic regression [23] were also introduced to improve the performance of classification tasks. QML, and specifically quantum deep learning, has more recent developments such as the QBM [24], the QCNN [25], and the QGANs [26], which generalize quantum machines to a variety of classification tasks. Quantum computation helped the author deliver anomaly detection services at LHC sites using a clustering method to enhance performance [27]. The application of QML by the author [28] resulted in cancer classification with the discovery of new biomarkers. A research study by the author applied genetic communities with tensor networks for optimizing QML in hepatocellular carcinoma classification [29]. Thus, these methods demonstrate how QML may go beyond to address the drawbacks of the classical ML.

In addition, there are various quantum-based classification approaches applied to the Iris dataset and recent researches related to Statlog HD dataset. As Iris is a benchmark dataset, it has been explored by various authors for classification using QML. The effectiveness of quantum kernel techniques in the initial learning phase was examined by the authors of [30], who especially investigated quantum

kernel-SVM (QK-SVM). They demonstrated the impact of various feature mappings on model performance by achieving an accuracy of 83.50% with the Y-FeatureMap and 84.50% with the ZFeatureMap. A kernel-based quantum classifier is one of the useful and influential QML methods for the hyperlinear classification of complex datasets. To demonstrate empirical subquadratic execution complexity with quantum operations, achievable even on NISQ processors, the authors of [31] presented the VQASVM algorithm. As a proof of concept, the algorithm was assessed on cloud-based NISQ machines using a toy dataset. To confirm its scalability and usefulness, they also conducted a numerical investigation of its performance on the Iris and MNIST datasets. The Iris classification accuracy obtained was 94.19% with $R = \infty$ and 95.34% with $R = 8192$.

The authors of [32] explored the application of QSVM on the NISQ architectures. They compared several quantum circuits, namely an optimized baseline and the HHL quantum circuit. Both methods resulted in a 97% accuracy, which proved that their method is resistant to quantum noise. The research work in [33] discusses improvements in QSVMs via variational kernel training. Their quantum kernel support vector machine (QK-SVM) attained an accuracy of 96.34% while the quantum variational SVM (QV-SVM) was at 95.43%. Further, their proposed quantum variational kernel-SVM (QVK-SVM) achieved a high accuracy of 98%. The authors of [34] propose a novel model in QML known as the quantum discriminator. A quantum-based model designed to handle binary classification tasks across datasets composed of binary features. Additionally, they introduce a hybrid quantum–classical algorithm for training the model, which operates with a time complexity of $O(N \log N)$. Further, they obtain 99% accuracy using the Iris dataset.

The recent study [35] related to HD prediction on Statlog dataset proposed GAPSO-RF, a new state-of-the-art framework that combines the GA and PSO algorithms with a RF model in order to improve feature selection and subsequently increase the efficiency of HD classification. The method begins with the help of multivariate statistical analysis to determine major features for the first population. A discriminate mutation strategy is then used in PSO for local search and in GA for global search with a specific concept of rehabilitation for individuals that have been eliminated during the process of selection. The GAPSO-RF model was applied on two available datasets, Cleveland and Statlog, achieving high prediction accuracy of 95.6% and 91.4%, respectively. In [36], the researchers carried out a predictive analysis to identify risk factors associated with HD using two publicly available datasets: Cleveland and Statlog. Classification models were created using the Cleveland dataset, and validation was performed on the Statlog dataset with 10-fold cross-validation. Among all the models employed, the SVM delivered the best accuracy and demonstrated comparable performance across both datasets. On the Statlog dataset, the accuracy of the SVM model was 87.04%.

Paper [37] suggested a novel approach of enhancing the interpretability of the HD prediction models, whereby a novel method known as extraction of classification rules from convolutional neural networks (ECRCNN) is used. To extract the interpretable rules, ECRCNN considers updated kernels inside the convolutional neural networks while providing better understanding of the factors affecting the heart disease. The method is tested on the Statlog (Heart) dataset with 85.19% accuracy. In

[38], the researcher came up with evolutionary-enhanced quantum supervised learning model which is tested on Iris dataset with 100% accuracy.

Increasing research interest in QML faces multiple obstacles despite its popularity. Research shows that most QML classification models have been tested on toy and synthetic datasets instead of real-world biomedical datasets. VQC circuits, along with many other quantum models that employ PQCs, encounter the barren plateau problem [39, 40], creating a vanishing gradient effect that hinders deep quantum model training. Quantum classifiers face restricted scalability [41] because of NISQ hardware constraints that prevent real-world datasets from being used in quantum models due to insufficient qubits in the current era. Quantum models struggle with interpretability problems that restrict their applicability to healthcare datasets that require transparency. Only a few researchers have utilized the advantages of hybrid quantum–classical models in combining classical performance with quantum feature transformation for classification tasks.

### 1.1 Motivation of the work

Operating in the NISQ era represents a barrier that limits the development of comprehensive quantum models because of severe hardware and scalability limitations. The combination of quantum kernels for feature transformation and LR for classification creates QEKLR, which serves as an operational and reliable substitute for deep quantum models restricted by the barren plateau problem. The shallow-depth ZZFeatureMap structure in QEKLR avoids training difficulties associated with the vanishing gradient problem when dealing with the barren plateau. QEKLR provides moderate interpretability by integrating LR, ensuring that classification decisions remain more transparent than deep quantum classifiers, which often function as black-box models. QEKLR solves NISQ hardware constraints through its operational mechanism with quantum kernels containing $\leq 4$ qubits, thereby enabling current quantum processor capabilities. Unlike prior QML studies that primarily validate models on toy synthetic datasets, QEKLR is applied to real-world datasets, including Statlog HD and Ecoli, demonstrating its applicability. This hybrid framework effectively bridges the gap between classical stability and quantum computational advantages, addressing the core limitations of existing QML models and paving the way for more efficient and interpretable quantum-enhanced classification systems.

### 1.2 Contribution of the research paper

The following is the contribution of the research work:

1. Performed classical preprocessing on each dataset includes outlier detection using IQR, feature extraction using PCA, and data normalization using MinMaxScaler.

2. Introduced a new method, QEKLR, which utilizes a shallow ZZFeatureMap circuit to encode classical data, generating a trainable quantum kernel through quantum fidelity for predictions in a logistic regression model.
3. To assess the performance of the QEKLR model, experiments were carried out on four benchmark datasets: the synthetic, Iris, Statlog HD, and Ecoli.
4. Comparative analysis of the performance of the QEKLR model with existing classical ML, QML, and recent classification approaches.

### 1.3 Organization Of remainder sections

The subsequent sections of this study are outlined as follows: Sect. 2 outlines the fundamentals of quantum computing. Section 3 discusses the preliminaries, and Sect. 4 details the proposed methodology, including the algorithm and performance metrics used for validation. Section 5 explains the datasets that were used in the research. Section 6 discusses the experimental setup and results. Finally, Sect. 7 concludes the research with future directions.

## 2 Quantum computing background

In this section, we have discussed the basics of quantum computing, the role of quantum gates (essential to build quantum circuits), and feature maps used in quantum computation.

### 2.1 Quantum computing fundamentals

QC leverages properties that are associated with quantum computing fundamental principles [42] like entanglement, superposition, and interference as illustrated in Fig. 1, making it significantly faster as compared to classical computing. In contrast with classical bits of information, a qubit, a different acronym for a quantum bit, can simultaneously exist in the $|0\rangle$ and $|1\rangle$ states. In QC, the principle of superposition states that qubits have the ability of existing in both states ($|0\rangle$ and $|1\rangle$) at once until measured.

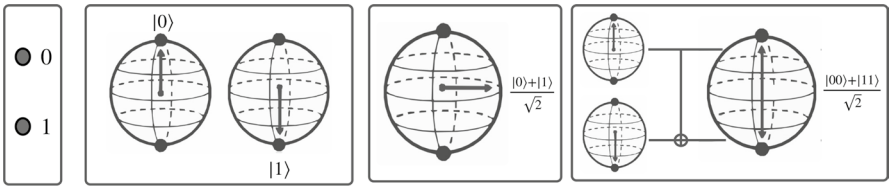



**Fig. 1** Quantum Computing Fundamentals

The superposition state may be expressed mathematically as $|\sigma\rangle = p|0\rangle + q|1\rangle$, given that qubits are represented as $|0\rangle$ and $|1\rangle$, $p$ and $q$ are complex numbers that correspond to $|p|^2 + |q|^2 = 1$. This property makes quantum computers capable of handling lots of data at the same time while offering an edge to particular operations. Entanglement, another quantum phenomenon, states that the state [43] of one qubit depends on the other, regardless of the distance between them. If two qubits are in the state $|\sigma\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$, then the two qubits are entangled, and as a result when one is measured, then the state of the other one is known. This phenomenon enables one to carry out several correlations and interconnections between different qubits, which is the fundamental of QC. Finally, quantum interference is defined based on the wave-like behavior of QS and the addition of probabilities, where the probability amplitudes interfere constructively or destructively. Through control of the interference patterns by using quantum gates in the quantum algorithms, the correct answers are strengthened while the wrong ones are made to cancel each other. This property is very important for more efficient quantum algorithms which include Grover's as well as Shor's algorithm.

### 2.2 Quantum gates

Quantum circuits, which are made up of qubits and quantum gates [44], are the fundamental units of quantum algorithms. A quantum gate acts on qubits changing their state via unitary operations. The Pauli-X, Y, and Z [45] are a few of the most often used quantum gates, which are one-qubit gates similar to classical NOT gates, used to reverse a qubit's state. The H gate that produces superposition states from classical binary states which is quite important in quantum parallelism. The CNOT gate, an elementary quantum gate, operates on two qubits and creates entanglement between them. Additionally, quantum gates and their operations are shown in Table 1.

**Table 1** Quantum Gates

| Quantum gate | Symbol | Operation |
|---|---|---|
| Pauli-X | $X$ | Similar to a traditional NOT gate, it flips a qubit's state |
| Pauli-Y | $Y$ | The Pauli-Y gate flips the qubit like the Pauli-X gate (i.e., it changes $\|0\rangle$ to $\|1\rangle$ and $\|1\rangle$ to $\|0\rangle$), but also introduces a quantum phase of -i |
| Pauli-Z | $Z$ | Applies a $-1$ phase to the $\|1\rangle$ state without altering $\|0\rangle$ |
| Hadamard | $H$ | It transforms a qubit from a definite state like $\|0\rangle$ or $\|1\rangle$ into a mixture of both states at once |
| CNOT | CNOT | If $\|1\rangle$ is the control qubit, then flips the target qubit |
| RX | $R_X(\theta)$ | Rotates a qubit around the X-axis by an angle of $\theta$ |
| RY | $R_Y(\theta)$ | Rotates a qubit around the Y-axis by an angle of $\theta$ |
| RZ | $R_Z(\theta)$ | Rotates a qubit around the Z-axis by an angle of $\theta$ |

These gates can help in making it possible to perform complex quantum algorithms through orienting the qubits in various ways such as creating superpositions, entangling qubits and changing their phases. In addition, the said operations are incorporated in the quantum circuits so that it can perform the more complex algorithms like Shor's and Grover's, which exhibit a faster problem-solving speed than the traditionally used classical algorithms. These gates lay the basis of the quantum computing because it allows one to control the qubits which make it easier to explore quantum and create the quantum technologies that could revolutionize fields like machine learning, cryptography, optimization, and material science.

### 2.3 Quantum feature maps

One essential component of QML is a quantum FM, which enables the conversion of classical data into QS such that quantum algorithms can be used for data manipulations. This encoding replaces classical inputs into quantum representations in a high-dimensional space by employing quantum gates and operations. Some of the FM's include: ZFeatureMap which involves rotation around the Bloch sphere's [46, 47] Z-axis; the ZZFeatureMap which includes entangling operations between qubits. Furthermore, the PauliFeatureMap uses the Pauli gates to develop a complex quantum model of the given input as shown in Table 2. These FM's utilize different quantum operations for encoding the data so as to utilize the quantum characteristics like superposition and entanglement and are effective in capturing higher-order patterns and interactions in data. The studies have identified that the selection and design of FM's are critical success factors of quantum classifiers and regressors.

## 3 Preliminaries

This preliminaries section provides an overview of the QML and ML algorithms that forms the conceptual basis of the proposed model.

### 3.1 VQC

VQC [16] is one of the QML algorithms, which operate with the help of a parameterized quantum circuit for classification. It begins with a dataset $D$ which has to undergo certain preprocessing to extract the features to be used in the quantum implementation. The received data are then subjected to FM $\Phi(x)$ whereby $x$

**Table 2** Quantum Feature Maps and Their Descriptions

| Feature Map | Description |
|---|---|
| ZFeatureMap | Applies Z-rotations to encode data |
| ZZFeatureMap | Adds Z-rotations + entanglement between qubits |
| PauliFeatureMap | Applies X-, Y-, Z-rotations and optional entanglement |

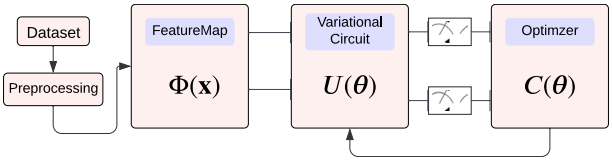
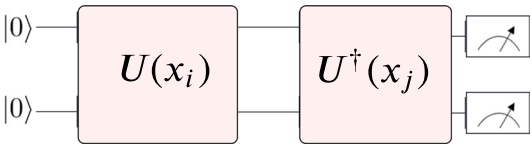



**Fig. 2** Variational Quantum Classifier

**Fig. 3** Quantum Kernel Estimation Circuit




represents the features, which convert the received data into the high-dimensional QS. This QS is then passed through a variational circuit, which we label as $U(\theta)$; $\theta$ are parameters that are amendable to training. In the variational circuit, the quantum state is altered or manipulated in such a way that results in its ability to produce a measurable value. The output measurements are employed in the calculation of a cost function $C(\theta)$ which defines the disparity between the forecasted labels and the true ones. Minimizing the function $C(\theta)$ with a classical optimizer is supposed to improve the ability of the quantum circuit to classify data correctly, by tuning the parameters $\theta$ as depicted in Fig. 2. This is an iterative method in that the parameters of the model are trained over and over again until they converge. The main advantage of the VQC here is in its capacity to utilize quantum entanglement and superposition thus giving optimal performance for specific tasks within the specific domain due to more complex decision boundaries. The quantum–classical integration of VQC establishes it as a highly effective tool in what is rapidly becoming the new frontier of QML.

### 3.2 QSVC

QSVC [14] extends the classical SVM by integrating QC principles. In this approach, the quantum kernel function captures complex relationships between data points in ways that classical kernels might not. The concept of quantum kernel methods can be understood by extending classical kernel methods. In quantum kernel methods [48–51], a classical data vector $x$ is mapped into a QS $|\phi(x)\rangle$ using an encoding circuit $U(x)$, which serves as the FM $\Phi(x)$. This quantum FM transforms classical data into the Hilbert space as depicted in Fig. 3. A quantum kernel function is then defined as the inner product of two QS, $|\phi(x_i)\rangle$ and $|\phi(x_j)\rangle$, in this space, expressed as $K = |\langle\phi(x_j)|\phi(x_i)\rangle|^2$. This can also be formulated as $|\langle 0^{\otimes N}|U^\dagger(x_j)U(x_i)|0^{\otimes N}\rangle|^2$.

Quantum kernel functions can be estimated by running the quantum circuit and measuring the probability of observing the output state $|0\rangle^{\otimes N}$. QSVC then applies a quantum version of the SVM optimization process to determine the best hyperplane

that separates classes. The classical optimization algorithm adjusts the cost parameters based on a cost function like classical SVM. By utilizing quantum entanglement and superposition, however, QSVC is able to learn finer decision boundaries which may improve classification performance on certain datasets.

### 3.3 Classical LR

LR [52] is a well-known algorithm among the statistical methods for binary classification tasks. It represents the likelihood that a given input sample is of a given class. The logistic regression model is defined as:

$$h(t) = \frac{1}{1 + e^{-t}}, \tag{1}$$

where $t$ represents a linear combination of the input features. For logistic regression, $t$ is calculated as

$$t = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_n x_n. \tag{2}$$

In this case, the intercept is $\theta_0$ and the coefficients are $\theta_i$ (for $i = 1, 2, \ldots, n$) for the predictor variables $x_i$. The logistic function, also called the sigmoid function, scales the input values and transforms them to probability values ranging between 0 and 1. The boundary is given by the weights, and to estimate these weights, the maximum likelihood estimation (MLE) technique is used. The cost function is given as follows:

$$L(\mathbf{v}) = -\sum_{j=1}^{M} \left[ z_j \log(Q(z_j \mid \mathbf{u}_j)) + (1 - z_j) \log(1 - Q(z_j \mid \mathbf{u}_j)) \right] \tag{3}$$

As shown in Eq. 3, the cost function should be minimal as it is the variation between the predicted and actual output. To determine the optimal weight values that result in the least cost function, optimization techniques like gradient descent are used.

### 3.4 KLR

KLR [53, 54] is an improvement over the basic LR as it employs the kernel trick to address non-linearity in input data. In classic LR, the probability of a class $y = 1$ given an input $x$ is expressed as:

$$P(y = 1|x) = \frac{1}{1 + e^{-g(x)}} \tag{4}$$

where the decision function $g(x)$ is defined as:

$$g(x) = v^T x + c \tag{5}$$

Here, $v^T x + c$ is a linear combination of the input features in the model. However, using a kernel function $\psi(x)$, KLR converts the input data into a higher-dimensional space. The decision-making function becomes:

$$g(x) = v^T \psi(x) + c \tag{6}$$

This transformation $\psi(x)$ allows for a nonlinear decision surface while still maintaining the probabilistic framework. Popular kernel functions used for this transformation include the RBF kernel and polynomial kernels. The parameters $v$ and $c$ are then optimized by finding the maximum likelihood of the model, similar to standard LR, but in the transformed feature space. This enables the model to capture the full non-linearity of the data while providing a probabilistic output for classification.

## 4 The proposed methodology

The following section explains in detail the proposed QEKLR approach for classification tasks. The QEKLR approach combines quantum computing with classical LR to improve classification. The classification using the proposed approach involves three stages as depicted in Fig. 4: classical preprocessing, applying the QEKLR model on the preprocessed dataset, and then the performance of the model assessed

**Table 3** List of Symbols

| Symbol notation | Description |
|---|---|
| $D$ | Dataset |
| $X$ | Classical data matrix |
| $X'$ | Transformed data matrix |
| $x$ | Input vector |
| $k$ | Number of classical features |
| $x_i$ | Individual data point (training feature vector) |
| $x_j$ | Individual data point (testing feature vector) |
| $y_i$ | Training class label |
| $y_j$ | Testing class label |
| $\phi(x)$ | Quantum state corresponding to input vector |
| $K(x_i, x_j)$ | Kernel function measuring inner product of QS |
| $z$ | Linear combination in logistic regression |
| $\beta_0$ | Intercept term in logistic regression |
| $\beta_i$ | Model parameters |
| $N_{\text{train}}$ | Number of training examples |
| $L(\beta)$ | Cross-entropy loss function |
| $\eta$ | Learning rate |
| $\hat{y}$ | Predicted class label |
| $\Sigma$ | Covariance matrix |
| $W$ | Eigenvector matrix in PCA |
| $T_P$ | True positive |
| $T_N$ | True negative |
| $F_P$ | False positive |
| $F_N$ | False negative |

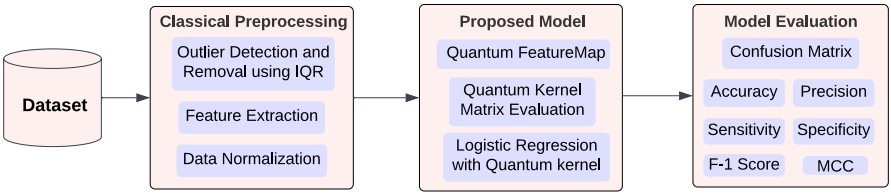



**Fig. 4** Proposed Methodology

using metrics $\text{Acc}_{\text{model}}, \text{Prec}_{\text{model}}, \text{Sen}_{\text{model}}, \text{Spec}_{\text{model}}, F1_{\text{model}}, \text{MCC}_{\text{model}}$. Symbols and their descriptions used in the methodology are shown in Table 3.

### 4.1 Classical preprocessing

This is the phase where preprocessing of the data takes place before the actual model training. The steps of data preprocessing may vary depending on the number of classes and certain features of the given dataset. When dealing with multiclass datasets, only two classes are selected. If the dataset is balanced, any two classes may be chosen. However, in the presence of class imbalance, selection is guided by the class distribution. When dealing with the nature of the dataset, it was assessed whether it contained categorical or numerical data, missing values, outliers, or noise and whether the data needed to be scaled or encoded. Suppose the dataset can be defined as $D = \{(x_i, y_i)\}_{i=1}^{N}$, where $x_i \in \mathbb{R}^n$ refers to the feature vectors and $y_i \in \{0, 1\}$ refers to the target variable. Initially, the detection and removal of outliers are done using the IQR. The IQR is determined by calculating the first quartile $Q1$ and third quartile $Q3$ for each feature. Outliers, on the other hand, are recognized as any values below $Q1 - 1.5 \times \text{IQR}$ or values above $Q3 + 1.5 \times \text{IQR}$. Outliers must be removed from the dataset because if the data contain some outliers, then such a model may not be very correct and may tend to mislead. Following outlier detection, PCA is applied to reduce the dimensionality of the dataset and extract the most important features. PCA is a technique that is used to compress the dataset while minimizing the amount of variance lost.

$$X' = XW \tag{7}$$

where $X$ represents the original data matrix, respectively; $W$ is the matrix of eigenvectors; and $X'$ represents the transformed data into the reduced feature space.

**Algorithm 1** Outlier Detection and Removal using IQR

**Require:** $D = \{(x_i, y_i)\}_{i=1}^{N}$
**Ensure:** $D$ without outliers
Initialize empty list $Outliers$
**for** each feature $x_j$ in $D$ **do**
  $Q1 \leftarrow \text{quantile}(x_j, 0.25)$
  $Q3 \leftarrow \text{quantile}(x_j, 0.75)$
  $\text{IQR} \leftarrow Q3 - Q1$
  $lower_{bound} \leftarrow Q1 - 1.5 \times \text{IQR}$
  $upper_{bound} \leftarrow Q3 + 1.5 \times \text{IQR}$
  **for** each $x_{ij}$ in $x_j$ **do**
    **if** $x_{ij} < lower_{bound}$ **or** $x_{ij} > upper_{bound}$ **then**
      Add $i$ to $Outliers$
    **end if**
  **end for**
**end for**
Remove entries with indices in $Outliers$ from $D$
**return** $D$

**Algorithm 2** Principal Component Analysis (PCA)

**Require:** Data matrix $X$
**Ensure:** Transformed data matrix $X'$
Center the data by subtracting the mean of each feature
covariance matrix $\Sigma = \frac{1}{N-1} X^T X$
eigenvalues and eigenvectors of $\Sigma$
Sort eigenvectors by decreasing eigenvalues
Select the top $k$ eigenvectors to form matrix $W$
$X' = XW$
**return** $X'$

**Algorithm 3** Data Normalization using MinMaxScaler

**Require:** Data matrix $X$
**Ensure:** Normalized data matrix $X'$
**for** each feature $x_j$ in $X$ **do**
  $min_j = \min(x_j)$
  $max_j = \max(x_j)$
  **for** each $x_{ij}$ in $x_j$ **do**
    $x_{ij} \Leftarrow \frac{x_{ij} - min_j}{max_j - min_j} \times (1 - (-1)) + (-1)$
  **end for**
**end for**
**return** $X'$

The matrix is applied in order to map the data into a low-dimensional space that preserves the variance of the data without much noise and redundancy. Lastly, the data normalization takes place with the help of MinMaxScaler, which scales the data in features so that it can lie on the equal scale. For every given feature, the minimum value and the maximum value are then determined and each individual value is normalized to the desired normalization scale in the range of -1 to 1. It scales each feature so as to compensate for features with large ranges as to not significantly affect the model. Thus, by applying these preprocessing techniques which include detection and rejection of outliers, feature reduction using PCA and feature normalization as discussed in Algorithm 1, 2 and 3, the given dataset is made stronger and more homogeneous and ready for quantum computing and machine learning. This is important especially for later use of the data in quantum machine learning as it deals with data quality and ensures proper alignment of the features for improved model performance.

### 4.2 The proposed model

The QEKLR proposed in this work combines the classical machine learning with quantum computing to take benefits of quantum feature map and quantum kernel method along with the classical logistic regression as shown in Fig. 5. The first stage involves Quantum feature mapping that encodes the classical data in a quantum feature space. For this purpose, we have employed the ZZFeatureMap, where the input vectors of dimension $k$(number of qubits, $k \leq 4$) are mapped into QS in the higher-dimensional Hilbert space, with two repetitions (repetition $\leq 2$) and full entanglement. ZZFeatureMap reformulates each classical data point as a quantum state so that quantum operations can be performed. The ZZFeatureMap was chosen for its ability to encode input data using entangling gates across qubits, which enhances the model's capacity to capture nonlinear feature interactions. Importantly, it provides configurable entanglement patterns (Linear, Circular, and Full), which allows us to experiment with different interaction complexities in a modular way. This flexibility is not directly supported in other standard feature maps like ZFeatureMap (which lacks entanglement) or PauliFeatureMap (which doesn't offer explicit entanglement structure control). This made ZZFeatureMap a practical and tunable choice in the NISQ context. Algorithm 4 describes the steps involved in encoding the classical data into QS.

After the quantum feature mapping, the model computes the quantum kernel matrix using the fidelity quantum kernel. This kernel matrix measures the similarity between QS by calculating the inner product of their corresponding quantum feature vectors. The resulting kernel values, which range between 0 and 1, provide a similarity measure that reflects how closely related two QS are in the high-dimensional quantum feature space. Unlike classical kernels, which utilize inner product computations in Euclidean space, quantum kernels utilize fidelity-based similarity measures that restrict their output values within a fixed range. The effectiveness of the quantum kernel, along with its interpretability, is enhanced due to its constrained values being both normalized and non-negative. Quantum kernels exhibit superior capability over conventional techniques in identifying complex nonlinear relationships through the integration of

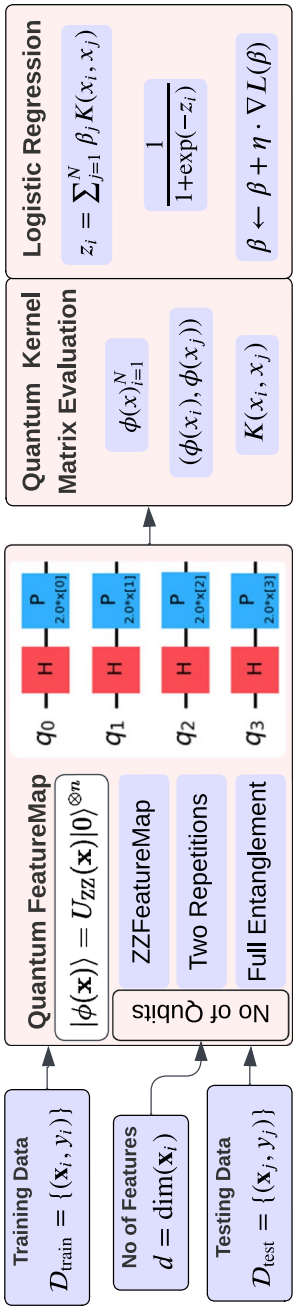



**Fig. 5** Proposed Model

quantum interference and entanglement phenomena. Since quantum states have perfect self-similarity qualities, the diagonal elements of the quantum kernel matrix are set to 1. Algorithm 5 describes the calculation of this kernel matrix.

**Algorithm 4** Quantum Feature Mapping using ZZFeatureMap

**Require:** Data matrix $X'$
**Ensure:** Quantum states $\phi(x)_{i=1}^{N}$
1: Initialize quantum circuit with $k$ qubits
2: **for** each sample $x_i'$ in $X'$ **do**
3: Apply ZZFeatureMap to $x_i'$ to get quantum state $\phi(x_i')$
4: **end for**
5: **return** $\phi(x)_{i=1}^{N}$

**Algorithm 5** Quantum Kernel Matrix Calculation

**Require:** Quantum states $\phi(x)_{i=1}^{N}$
**Ensure:** Quantum kernel matrix $K$
1: **for** each pair of samples $(\phi(x_i), \phi(x_j))$ **do**
2: Compute $K(x_i, x_j) = |\langle\phi(x_i)|\phi(x_j)\rangle|^2$
3: **end for**
4: **return** $K$

**Algorithm 6** Logistic Regression with Quantum Kernel

**Require:** Quantum kernel matrix $K(x_i, x_j)$, labels $y$
**Ensure:** Model parameters $\beta$
1: Decision Boundary $z_i = \sum_{j=1}^{N} \beta K(x_i, x_j)$
2: Initialize parameters $\beta$ to small random values
3: **while** not converged **do**
4: Compute predictions $P(y = 1|x_i) = \frac{1}{1+\exp(-z_i)}$
5: Compute gradient $\nabla L(\beta)$
6: Update parameters $\beta \Leftarrow \beta - \eta \nabla L(\beta)$
7: **end while**
8: **return** $\beta$

The final integration of quantum computations with classical methods occurs in the logistic regression with quantum kernel phase. Here, the quantum kernel matrix is incorporated into the logistic regression model. Instead of using a linear combination of classical features, the model utilizes the kernel values to capture complex patterns and relationships. The logistic regression model computes the probability of class membership using a logistic function applied to the kernel values. The

model parameters are optimized through gradient descent, which iteratively adjusts the parameters to minimize the cross-entropy loss function. Algorithm 6 details this process, including the computation of predictions, the calculation of gradients, and the parameter updates until convergence is achieved.

### 4.3 Model evaluation

In the prediction and model evaluation phase, new data points are transformed into the quantum feature space using the same FM applied during training. The kernel values of these new data points are calculated relative to the training data, and class probabilities are estimated using a logistic regression model. Specifically, for each sample $x_i$ in the quantum kernel matrix $K_{\text{test}}$, the algorithm computes $z_i = \beta_0 + \sum_{j=1}^{N_{\text{train}}} \beta_j K(x_i, x_j)$, where $\beta$ represents the model parameters and the prediction $\hat{y}_i$ is determined by the sign of $z_i$ as detailed in Algorithm 7. This process yields predicted labels $\hat{y}$. To evaluate the model's performance comprehensively, various metrics are computed. The evaluation includes calculating $\text{Acc}_{\text{model}}, \text{Prec}_{\text{model}}, \text{Sen}_{\text{model}}, \text{Spec}_{\text{model}}, F1_{\text{model}}, \text{MCC}_{\text{model}}$ based on the $\text{CM}_{\text{model}}$, which consists of $T_P, T_N, F_P$ and $F_N$. The performance metrics provide a thorough assessment of the model's effectiveness, as detailed in Algorithm 8, which outlines the process used for these calculations.

**Algorithm 7** Prediction using Trained Logistic Regression Model

**Require:** Model parameters $\beta$, quantum kernel matrix $K_{\text{test}}$
**Ensure:** Predicted labels $\hat{y}$
1: **for** each sample $x_i$ in $K_{\text{test}}$ **do**
2: Compute $z_i = \beta_0 + \sum_{j=1}^{N_{\text{train}}} \beta_j K(x_i, x_j)$
3: Compute prediction $\hat{y}_i = \text{sign}(z_i)$
4: **end for**
5: **return** $\hat{y}$

**Algorithm 8** Model Evaluation Metrics

**Require:** True labels $y$, predicted labels $\hat{y}$
**Ensure:** $\text{Acc}_{\text{model}}, \text{Prec}_{\text{model}}, \text{Sen}_{\text{model}}, \text{Spec}_{\text{model}}, F1_{\text{model}}, \text{MCC}_{\text{model}}$
1: Compute $\text{CM}_{\text{model}}$: $T_P$, $T_N$, $F_P$, $F_N$
2: Compute $\text{Acc}_{\text{model}} = \frac{T_P + T_N}{T_P + T_N + F_P + F_N}$
3: Compute $\text{Prec}_{\text{model}} = \frac{T_P}{T_P + F_P}$
4: Compute $\text{Sen}_{\text{model}} = \frac{T_P}{T_P + F_N}$
5: Compute $\text{Spec}_{\text{model}} = \frac{T_N}{T_N + F_P}$
6: Compute $F1_{\text{model}} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$
7: Compute $\text{MCC}_{\text{model}} = \frac{(T_P \times T_N) - (F_P \times F_N)}{\sqrt{(T_P + F_N)(T_P + F_P)(T_N + F_N)(T_N + F_P)}}$
8: **return** $\text{Acc}_{\text{model}}, \text{Prec}_{\text{model}}, \text{Sen}_{\text{model}}, \text{Spec}_{\text{model}}, F1_{\text{model}}, \text{MCC}_{\text{model}}$

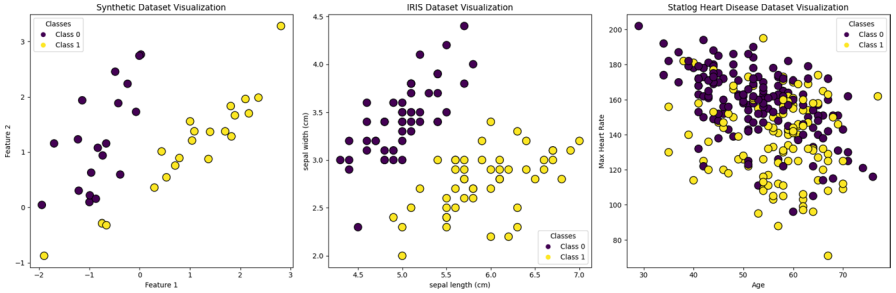



**Fig. 6** Synthetic, Iris, and Statlog HD Dataset Visualization

**Table 4** Dataset Description

| Dataset | No. of classes | No. of features | Total size |
|---|---|---|---|
| Synthetic | 2 | 2 | 40 |
| Iris | 2 | 4 | 100 |
| Statlog HD | 2 | 13 | 270 |
| Ecoli | 8 | 7 | 336 |

## 5 Datasets description

An overview of all the datasets utilized to conduct the experiments is given in this section.

### 5.1 Dataset 1: synthetic dataset

A synthetic dataset [55] was generated randomly to perform binary classification tasks. The dataset consists of 40 samples and each sample consists of 2 features. There were no missing values and outliers present in the dataset. The dataset was created to assess the performance of the QEKLR model.

### 5.2 Dataset 2: Iris dataset

The Iris dataset [56], used for our binary classification experiments, focuses on two classes: the Iris Setosa which is classified as '0' and the Iris Versicolor which is classified as '1.' It contains 100 instances, each with four continuous features: They are the inputs for the model or the independent variables, namely sepal length, sepal width, petal length, and petal width. No missing values and outliers were present in the dataset. This subset enables the accurate development and comparison of classification models.

### 5.3 Dataset 3: Statlog HD dataset

Statlog HD dataset [57] is used to perform binary classification to classify patients with HD. It contains 270 instances and 13 attributes like age, sex, chest parameters, cholesterol levels, blood sugar level, heart rate, and types of chest pain. The dependent variable represents the presence of HD; if it is present, the value '1' is assigned, else '0' as depicted in Fig. 6. In practice, this dataset assists in generating models to forecast the likelihood of patients developing HDs. Table 4 describes the summary of each dataset.

### 5.4 Dataset 4: Ecoli dataset

The Ecoli dataset [58] contains 336 instances with 7 continuous features and 8 different classes, used for multiclass classification to predict protein localization sites in Escherichia coli. It includes attributes like signal sequence scores, amino acid composition, and membrane-spanning predictions and it is widely used in biomedical research.

## 6 Results and discussion

These experiments were conducted in a controlled environment using Qiskit 1.0 SDK, Qiskit Machine Learning 0.7.2, Qiskit Algorithms 0.3.0, and Python 3.9. Qiskit SDK, a framework for QC and related applications that enables users to create and simulate quantum algorithms, perform QML tasks, and analyze the results. To evaluate the model performance, firstly the experiments were performed for 2 features on the synthetic dataset. The synthetic dataset generated contained no missing values and outliers. The total number of instances in the synthetic and Iris datasets stayed the same after applying interquartile range (IQR) filtering. However, for the Statlog dataset, the number of instances changed from 270 to 192, and for the Ecoli dataset, it changed from 336 to 306. After this, each dataset was split into 80% for training and 20% for testing. Classical ML classifiers applied on the dataset were RF, LR, CART, KLR, and QML classifiers applied

**Table 5** ML and QML Model Performance Metrics for Synthetic Dataset Classification 2 Features

| Model | $Acc_{model}$ | $Prec_{model}$ | $Sen_{model}$ | $Spec_{model}$ | $F1_{model}$ | $MCC_{model}$ |
|---|---|---|---|---|---|---|
| RF | 100.0 | 100.0 | 100.0 | 100.0 | 100.0 | 1.0 |
| LR | 100.0 | 100.0 | 100.0 | 100.0 | 100.0 | 1.0 |
| CART | 100.0 | 100.0 | 100.0 | 100.0 | 100.0 | 1.0 |
| KLR | 100.0 | 100.0 | 100.0 | 100.0 | 100.0 | 1.0 |
| VQC | 95.00 | 88.89 | 100.0 | 91.67 | 94.12 | 0.9 |
| QSVC | 100.0 | 100.0 | 100.0 | 100.0 | 100.0 | 1.0 |
| **QEKLR(Proposed)** | 100.0 | 100.0 | 100.0 | 100.0 | 100.0 | 1.0 |

**Table 6** ML and QML Model Performance Metrics for Iris Dataset Classification 4 Features

| Model | $\text{Acc}_{\text{model}}$ | $\text{Prec}_{\text{model}}$ | $\text{Sen}_{\text{model}}$ | $\text{Spec}_{\text{model}}$ | $\text{F1}_{\text{model}}$ | $\text{MCC}_{\text{model}}$ |
|---|---|---|---|---|---|---|
| RF | 100.0 | 100.0 | 100.0 | 100.0 | 100.0 | 1.0 |
| LR | 100.0 | 100.0 | 100.0 | 100.0 | 100.0 | 1.0 |
| CART | 100.0 | 100.0 | 100.0 | 100.0 | 100.0 | 1.0 |
| KLR | 100.0 | 100.0 | 100.0 | 100.0 | 100.0 | 1.0 |
| VQC | 90.00 | 90.00 | 92.00 | 83.00 | 90.00 | 0.8 |
| QSVC | 100.0 | 100.0 | 100.0 | 100.0 | 100.0 | 1.0 |
| **QEKLR(Proposed)** | 100.0 | 100.0 | 100.0 | 100.0 | 100.0 | 1.0 |

on the dataset were VQC, QSVC, and proposed QEKLR. The performance of the ML and QML classifiers on the synthetic dataset is shown in Table 5 using metrics $\text{Acc}_{\text{model}}, \text{Prec}_{\text{model}}, \text{Sen}_{\text{model}}, \text{Spec}_{\text{model}}, F1_{\text{model}}, \text{MCC}_{\text{model}}$.

Similarly, we performed experiments on the benchmark Iris dataset for 4 features. This benchmark dataset contains 4 features; hence, no PCA is applied to it. We classified the Iris dataset using the same ML and QML classifiers that we used to classify the synthetic dataset. Table 6 displays the performance of each model on the Iris dataset using the same performance metrics. Several models achieved 100% accuracy. This outcome can be attributed to the inherent characteristics of the dataset, which consists of two distinct and linearly separable classes. Such a structure significantly simplifies the classification task, allowing models to perform with high precision.

We employed the Statlog HD dataset for classification after evaluating the performance of our proposed QEKLR model on synthetic and Iris datasets for 2 features and 4 features, respectively. Firstly, the dataset was preprocessed as discussed in Algorithm 1, 2 and 3. The dataset was subsequently divided into training and testing sets using an 80:20 ratio. To classify the dataset using the proposed QEKLR model, the data first need to be converted into quantum states to prepare it for quantum systems. In the training set, the ZZFeatureMap was employed for mapping of classical data into quantum states as represented in Fig. 7. Figure 8 depicts the result of calculating the quantum kernel matrix for the training and test datasets following

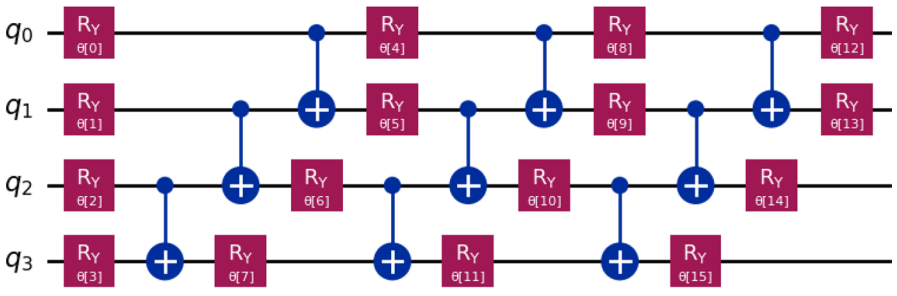
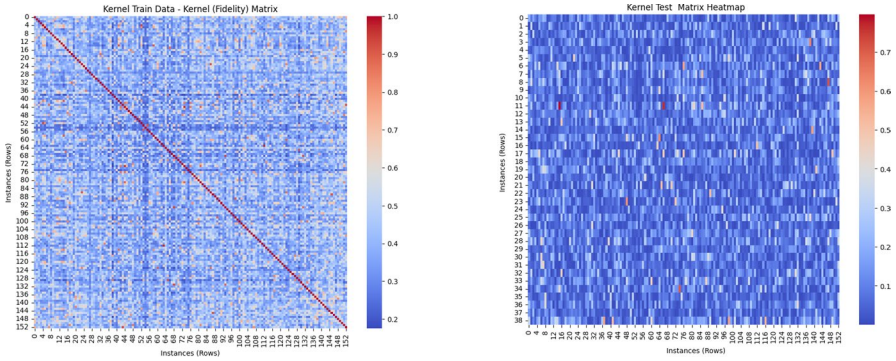
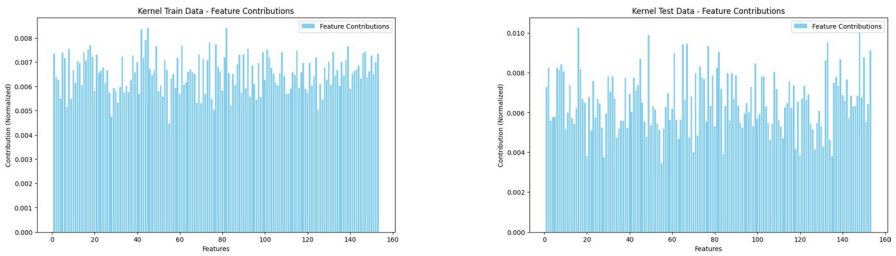



**Fig. 7** ZZFeatureMap of 4-Qubits




(a) Kernel Train Matrix (b) Kernel Test Matrix

**Fig. 8** Kernel Fidelity Matrix for the Proposed Model



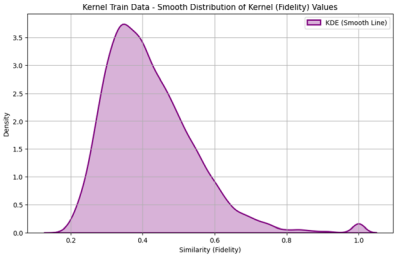

(a) Kernel Train-Data Feature Contributions (b) Kernel Test-Data Feature Contributions

**Fig. 9** Kernel Train–Test Data Feature Contributions for the Proposed Model

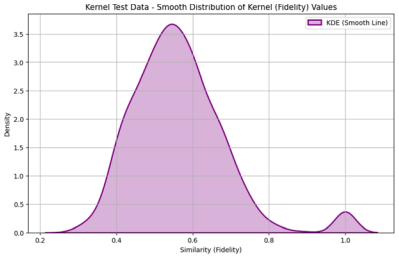
a quantum-enhanced kernel approach. More specifically, the kernel matrix is computed with ZZFeatureMap that works jointly with the fidelity quantum kernel strategy. This ZZFeatureMap changes the original data into a quantum feature space making it possible to study interactions between features that cannot be studied by the classical kernels. This method makes use of the fact that QC is able to solve for features in high dimensionality, thus improving the efficacy of the proposed QEKLR model in partitioning and classifying complex datasets.

The important details of the data include the normalized feature contributions which help to indicate how much contribution each of the features makes to the corresponding classification model as depicted in Fig. 9. The contribution of the features is shown with the help of a bar graph which indicates that there are no significant variations in which one feature has highly influenced than the other one; nearly equal balanced variations have been found for the training as well as for the test dataset. This balance makes the kernel method distribute the amount of weight of the features equally hence making it sensitive to overfit on certain features.

Moreover, the kernel fidelity matrix for both training and test data illustrates the pairwise similarities between data instances, with values ranging from 0 (no




(a) Smooth Distribution of Kernel Train Data Values




(b) Smooth Distribution of Kernel Test Data Values

**Fig. 10** Kernel Fidelity Distribution

similarity) to 1 (perfect similarity). These matrices show strong diagonal dominance, as expected, since each data point is most similar to itself. Off-diagonal elements reveal the similarity between different instances, with the color gradient from red (low fidelity) to red (high fidelity) indicating varying degrees of similarity. Such visual representations are essential for understanding the relationships within the data and how the quantum kernel captures these.

Moreover, kernel fidelity distribution is shown in a kernel density estimation (KDE) plot in Fig. 10 to understand how dispersed the values of similarity are and how condensed they are. As seen from the KDE plots, the majority of the points have moderate similarity values with the maximum density around 0.4 to 0.6. On the other hand, we can also observe a few cases where similarity is very high, almost equal to 1.0, which means that few data points have almost similar feature representations in the quantum space. When machine learning models are being trained, these distinctions in similarity representation facilitate more

**Table 7** Comparative Analysis of Baseline ML Models with Outliers for Statlog HD Dataset Classification 13 Features

| Model | $Acc_{model}$ | $Prec_{model}$ | $Sen_{model}$ | $Spec_{model}$ | $F1_{model}$ | $MCC_{model}$ |
|---|---|---|---|---|---|---|
| RF | 85.19 | 84.21 | 76.19 | 90.91 | 80.00 | 0.69 |
| LR | 87.04 | 79.17 | 90.48 | 84.85 | 84.44 | 0.74 |
| CART | 68.52 | 57.14 | 76.19 | 63.64 | 65.31 | 0.39 |
| KLR | 85.19 | 78.26 | 85.71 | 84.85 | 81.82 | 0.70 |

**Table 8** Comparative Analysis of Baseline ML Models without Outliers for Statlog HD Dataset Classification 13 Features

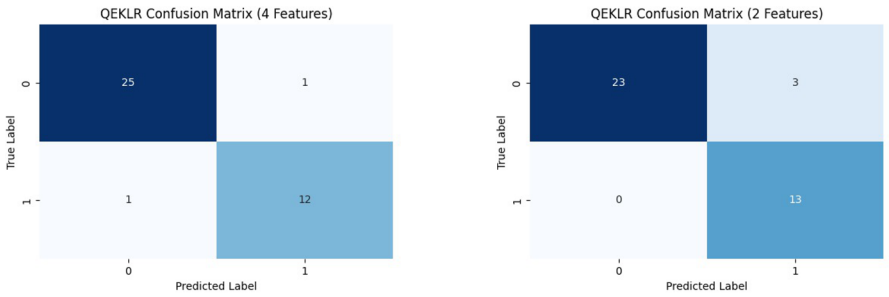
| Model | $Acc_{model}$ | $Prec_{model}$ | $Sen_{model}$ | $Spec_{model}$ | $F1_{model}$ | $MCC_{model}$ |
|---|---|---|---|---|---|---|
| RF | 87.18 | 75.00 | 92.31 | 84.62 | 82.76 | 0.74 |
| LR | 84.62 | 73.33 | 84.62 | 84.62 | 78.57 | 0.67 |
| CART | 79.49 | 69.23 | 69.23 | 84.62 | 69.23 | 0.54 |
| KLR | 87.18 | 75.00 | 92.31 | 84.62 | 82.76 | 0.74 |

**Table 9** ML and QML Model Performance Metrics for Statlog HD Dataset Classification 2 Features

| Model | $Acc_{model}$ | $Prec_{model}$ | $Sen_{model}$ | $Spec_{model}$ | $F1_{model}$ | $MCC_{model}$ |
|---|---|---|---|---|---|---|
| RF | 84.62 | 73.33 | 84.62 | 84.62 | 78.57 | 0.67 |
| LR | 89.74 | 84.62 | 84.62 | 92.31 | 84.62 | 0.77 |
| CART | 82.05 | 71.43 | 76.92 | 84.62 | 74.07 | 0.60 |
| KLR | 79.49 | 66.67 | 76.92 | 80.77 | 71.43 | 0.56 |
| VQC | 62.00 | 50.00 | 50.00 | 85.00 | 48.00 | 0.00 |
| QSVC | 72.00 | 68.00 | 63.00 | 88.00 | 64.00 | 0.31 |
| QEKLR (Proposed 1) | 92.31 | 81.25 | 100.0 | 88.46 | 89.65 | 0.85 |

**Table 10** ML and QML Model Performance Metrics for Statlog HD Dataset Classification 4 Features

| Model | $Acc_{model}$ | $Prec_{model}$ | $Sen_{model}$ | $Spec_{model}$ | $F1_{model}$ | $MCC_{model}$ |
|---|---|---|---|---|---|---|
| RF | 87.18 | 78.57 | 84.62 | 88.46 | 81.48 | 0.72 |
| LR | 92.31 | 85.71 | 92.31 | 92.31 | 88.88 | 0.83 |
| CART | 84.62 | 73.33 | 84.62 | 84.62 | 78.57 | 0.67 |
| KLR | 92.31 | 81.25 | 100.0 | 88.46 | 89.66 | 0.85 |
| VQC | 59.00 | 54.00 | 54.00 | 69.00 | 54.00 | 0.08 |
| QSVC | 74.00 | 72.00 | 67.00 | 88.00 | 68.00 | 0.39 |
| QEKLR(Proposed 2) | 94.87 | 92.31 | 92.31 | 96.15 | 92.31 | 0.88 |

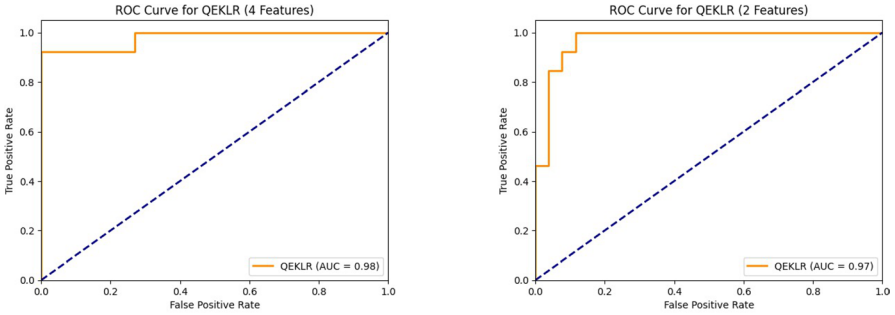



(a) QEKLR (Proposed 2) (b) QEKLR (Proposed 1)

**Fig. 11** Confusion Matrix of QEKLR (Proposed 1 and Proposed 2)

generalization, especially in tasks that involve the categorization of almost similar instances.

Tables 7 and 8 represent a comparative analysis of baseline ML models such as RF, LR, CART, and KLR on the Statlog HD for all 13 features, both with and without outliers. Additionally, the same ML classifiers were applied to the Statlog HD dataset to evaluate the effectiveness of the model for 2 and 4 features, respectively. The QML classifiers were also applied on the Statlog HD dataset,




(a) QEKLR (Proposed 2) (b) QEKLR (Proposed 1)

**Fig. 12** ROC Curve for the Proposed Model

**Table 11** ML and QML Model Performance Metrics for Ecoli Dataset Classification 3 Features

| Model | $Acc_{model}$ | $Prec_{model}$ | $Sen_{model}$ | $Spec_{model}$ | $F1_{model}$ | $MCC_{model}$ |
|---|---|---|---|---|---|---|
| RF | 70.97 | 72.99 | 35.76 | 35.76 | 33.59 | 57.76 |
| LR | 66.13 | 60.40 | 25.32 | 25.32 | 23.54 | 49.56 |
| CART | 62.90 | 40.63 | 26.75 | 26.75 | 27.05 | 46.74 |
| QEKLR | 70.97 | 72.48 | 31.62 | 31.62 | 31.16 | 58.14 |

including VQC, QSVC, and the proposed QEKLR model. The performance analysis of ML and QML classifiers on Statlog HD dataset for 2 and 4 features is shown in Tables 9 and 10 respectively.

The experimental findings demonstrate that our proposed QEKLR model (Proposed 1 and Proposed 2) outperforms the performance of both classical (RF, LR, CART, KLR) and quantum (VQC, QSVC) ML models on the Statlog dataset, achieving 92.31% accuracy with 2 features and 94.87% accuracy with 4 features. QEKLR (Proposed 2) achieves an MCC of 0.88, which is higher than all models such as RF(0.72), LR(0.83), CART(0.67), KLR(0.85) and QEKLR (Proposed 1) (0.85). MCC is a crucial metric that takes into account every aspect of the CM ($T_P$, $T_N$, $F_P$, $F_N$) as depicted in Fig. 11. This makes it especially useful for assessing models on imbalanced datasets. MCC of the QEKLR (Proposed 2) is relatively higher compared to the other models, which implies more reliability of the classifier with less misclassification across all the classes. Furthermore, Proposed 2 has better overall performance with an excellent balance between specificity (96.15%) and accuracy (94.87%). Proposed 1 attains 100% sensitivity, but its marginally reduced MCC raises the possibility of trade-offs in other domains, namely specificity (88.46%). Conventional models such as LR are competitive in terms of accuracy (92.31%) and MCC (0.83), but they fall short of the QEKLR models' robustness and consistency.

In addition to the accuracy and MCC, the AUC-ROC metrics show the enhanced classification performance of the QEKLR models (Proposed 1 and Proposed 2). AUC-ROC is an essential metric for assessing a model's ability to distinguish between

**Table 12** Comparative Analysis of the Recent Methods on the Iris Dataset

| Reference | Model | Accuracy |
|---|---|---|
| [30] | Qk-SVM(Z-feature Map) | 84.50 |
| [31] | VQASVM | 94.19 |
| [32] | QSVM with HHL circuit | 97.00 |
| [33] | QVK-SVM | 98.00 |
| [34] | Quantum Discriminator model | 99.00 |
| [38] | EEQSL | 100.0 |
| QEKLR (Proposed) | QEKLR | 100.0 |

classes. AUC-ROC values of 0.97 for QEKLR (Proposed 1) and 0.98 for QEKLR (Proposed 2) were obtained as depicted in Fig. 12, in which QEKLR (Proposed 2) is marginally superior to QEKLR (Proposed 1). These high statistics show that the instances can be classified almost perfectly by the QEKLR (Proposed 2) Model.

We proceeded to apply the QEKLR model on the Ecoli dataset for additional performance evaluation following the Statlog HD examination. This data set required the implementation of the QEKLR model with three features. The QEKLR reached 70.97% accuracy that matched the RF model while demonstrating superior accuracy than LR and CART as shown in Table 11.

The main difficulty in classifying Ecoli data over Statlog Heart Disease data appears through both complex distribution patterns and challenging class characteristics. Model classification becomes more straightforward with the Statlog dataset because it includes only two classes. Model accuracy for Ecoli depends significantly on how its multiple categorical classes get converted to binary classification.

A thorough comparison of the suggested QEKLR model with several cutting-edge techniques using the Iris and Statlog datasets is shown in Tables 12 and 13, respectively. The proposed QEKLR model demonstrated its superiority in classification tasks by achieving an unparalleled 100% accuracy on the Iris dataset (Table 12), outperforming other methods such as the Quantum Discriminator Model (99%) and QVK-SVM (98%). Likewise, Table 13 presents QEKLR's performance on the Statlog dataset. A comparison of QEKLR (Proposed 1 and Proposed 2) with recent models such as SVM (87.04%), ECRCNN (85.19%), and GAPSO-RF (91.40%) shows a notable improvement in accuracy, with QEKLR (Proposed 1) achieving 92.31% and QEKLR (Proposed 2) reaching 94.87%.

A comparative study of many ML and QML models' performance on the Statlog HD dataset is presented in Fig. 13. Using 4 features, the QEKLR (Proposed 2)

**Table 13** Comparative Analysis of the Recent Methods on the Statlog Dataset

| Reference | Model | No. of Features | Accuracy |
|---|---|---|---|
| [35] | GAPSO-RF | 9 | 91.40 |
| [36] | SVM | 13 | 87.04 |
| [37] | ECRCNN | 13 | 85.19 |
| QEKLR (Proposed 1) | QEKLR | 2 | 92.31 |
| QEKLR (Proposed 2) | QEKLR | 4 | 94.87 |

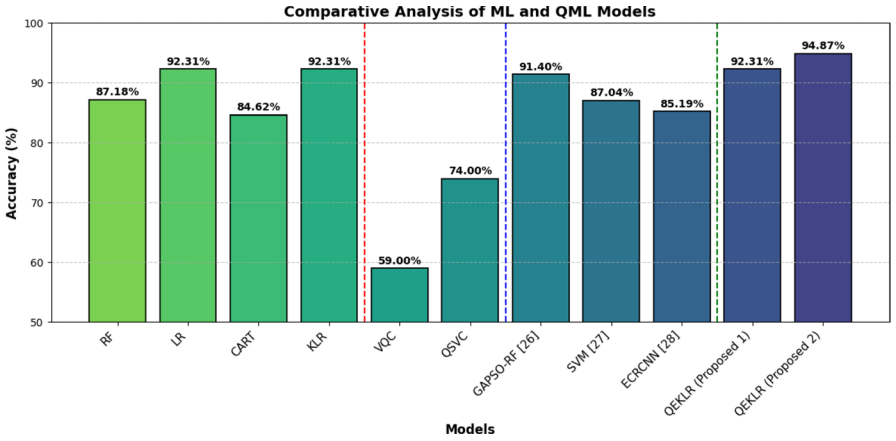



**Fig. 13** $Acc_{model}$ Comparison of the Proposed QEKLR Model with Baseline Models and Recent Models

outperforms all other models with an accuracy of 94.87%. The QEKLR (Proposed 1) achieves an accuracy of 92.31% using 2 features, demonstrating strong competitiveness. These results demonstrate the efficacy of the QEKLR method by surpassing the performance of cutting-edge models like GAPSO-RF (91.40%), SVM (87.04%), and ECRCNN (85.19%). While RF, CART achieve 87.18% and 84.62%, respectively, among the baseline models, LR and KLR both score 92.31% accuracy with 4 features. Overall, this quantum kernel-based approach, enabled by advanced quantum feature mapping, significantly improves the model's capacity to uncover intricate relationships within the data. The resulting kernel matrix serves as a powerful tool for feature evaluation, similarity detection, and classification, especially in datasets where classical methods may struggle to fully capture the underlying patterns. By visualizing these data through feature contributions, kernel fidelity matrices, and KDE plots, we can gain deep insights into the effectiveness of quantum-enhanced techniques in handling complex, nonlinear data structures.

## 7 Conclusion and future scope

This research introduced a novel hybrid approach QEKLR, which utilizes quantum kernel with LR to enhance classification. The model was tested on two benchmark datasets (synthetic and Iris) and two real-world datasets (Statlog HD and Ecoli) demonstrating superior accuracy with 94.87% on Statlog HD dataset; for synthetic, Iris, and Ecoli it shows the comparable performance to classical ML models suggesting dataset-dependent behavior. QEKLR avoids the barren plateau issue commonly observed in PQC by employing shallow-depth feature map-based quantum kernel, which leverages fidelity-based distance metrics rather than a trainable variational circuit. In order to ensure efficiency for both current quantum devices and simulators, QEKLR uses a shallow-depth feature map with a maximum of 4 qubits and 2 repetitions in the current NISQ era. In addition, QEKLR enhances interpretability

over quantum models based on PQCs, by integrating the quantum kernel with logistic regression (LR). While maintaining the ability to examine the decision boundaries of the logistic regression model for predictive analysis, this hybrid strategy ensures a clear understanding of the quantum kernel's role in data transformation while QEKLR offers a practical and effective approach for classification tasks in NISQ-compatible machine learning applications by striking a balance between quantum computational advantages and classical interpretability, but as dataset sizes and qubit counts increase, quantum kernel techniques encounter scalability challenge. One major issue is kernel stability, where larger datasets lead to exponential concentration, causing kernel values to cluster within a narrow range. This reduces their ability to distinguish between different data points, weakening classification accuracy. Future research will concentrate on improving quantum kernel techniques for medium- to large-scale datasets by including hybrid quantum–classical optimization, more effective feature maps, and quantum transfer learning (QTL) to increase their adaptability.

**Author contributions** Both authors contributed equally to this work.

**Funding** No external funding provided.

**Data availability** The datasets used in this study are publicly accessible and can be obtained from https://github.com/shreshthamisra03/SytheticData_2_features, https://archive.ics.uci.edu/dataset/53/iris, https://archive.ics.uci.edu/dataset/145/statlog+heart, and https://archive.ics.uci.edu/dataset/39/ecoli.

## Declarations

**Conflict of interest** The authors declare that they have no conflict of interest.

## References

1. Bishop CM (2006) Pattern Recognition and Machine Learning. Springer, New York. https://link.springer.com/book/9780387310732
2. Sarker IH (2021) Machine learning: Algorithms, real-world applications and research directions. SN Comput Sci 2(3):160. https://doi.org/10.1007/s42979-021-00592-x. (**Epub 2021 Mar 22**)
3. Zhang S, Li J (2023) Knn classification with one-step computation. IEEE Trans Knowl Data Eng 35(3):2711–2723. https://doi.org/10.1109/TKDE.2021.3119140
4. Zhang S, Li J, Li Y (2023) Reachable distance function for knn classification. IEEE Trans Knowl Data Eng 35(7):7382–7396. https://doi.org/10.1109/TKDE.2022.3185149
5. Biamonte J, Wittek P, Pancotti N, Rebentrost P, Wiebe N, Lloyd S (2017) Quantum machine learning. Nature 549(7671):195–202. https://doi.org/10.1038/nature23474
6. Zhang S, Li J, Zhang W, Qin Y (2022) Hyper-class representation of data. Neurocomputing 503:200–218. https://doi.org/10.1016/j.neucom.2022.06.082
7. Nielsen MA, Chuang IL (2010) Quantum Computation and Quantum Information: 10th Anniversary Edition. Cambridge University Press, New York. https://books.google.com/books?id=-s4DEy7o-a0C
8. Qi H, Xiao S, Liu Z et al (2024) Variational quantum algorithms: fundamental concepts, applications and challenges. Quantum Inf Process 23:224. https://doi.org/10.1007/s11128-024-04438-2
9. Preskill J (2018) Quantum computing in the nisq era and beyond. Quantum 2:79

10. Shor PW (1997) Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. SIAM J Comput 26(5):1484–1509. https://doi.org/10.1137/S0097539795293172
11. Grover LK (1996) A fast quantum mechanical algorithm for database search. In: Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing, pp. 212–219. Association for Computing Machinery, New York, NY, USA. https://doi.org/10.1145/237814.237866
12. Harrow AW, Hassidim A, Lloyd S (2009) Quantum algorithm for linear systems of equations. Phys Rev Lett 103:150502. https://doi.org/10.1103/PhysRevLett.103.150502
13. Anguita D, Ridella S, Rivieccio F, Zunino R (2003) Quantum optimization for training support vector machines. Neural Netw 16(5–6):763–770. https://doi.org/10.1016/S0893-6080(03)00087-X
14. Rebentrost P, Mohseni M, Lloyd S (2014) Quantum support vector machine for big data classification. Phys Rev Lett 113:130503. https://doi.org/10.1103/PhysRevLett.113.130503
15. Li J, Li Y, Song J, Zhang J, Zhang S (2024) Quantum support vector machine for classifying noisy data. IEEE Trans Comput 73(9):2233–2247. https://doi.org/10.1109/TC.2024.3416619
16. Maheshwari D, Sierra-Sosa D, Garcia-Zapirain B (2022) Variational quantum classifier for binary classification: Real vs synthetic dataset. IEEE Access 10:3705–3715. https://doi.org/10.1109/ACCESS.2021.3139323
17. Wiebe N, Kapoor A, Svore KM (2015) Quantum algorithms for nearest-neighbor methods for supervised and unsupervised learning. Quantum Inf Comput 15:316–356. https://doi.org/10.5555/2871393.2871400
18. Li J, Zhang J, Zhang J, Zhang S (2024) Quantum knn classification with k value selection and neighbor selection. IEEE Trans Comput-Aided Des Integr Circuits Syst 43(5):1332–1345. https://doi.org/10.1109/TCAD.2023.3345251
19. Heese R, Bickert P, Niederle AE (2022) Representation of binary classification trees with binary features by quantum circuits. Quantum 6:676. https://doi.org/10.22331/q-2022-03-30-676
20. Lu S, Braunstein SL (2014) Quantum decision tree classifier. Quantum Inf Process 13:757–770. https://doi.org/10.1007/s11128-013-0687-5
21. Khadiev KR, Safina L (2021) The quantum version of random forest model for binary classification problem. https://api.semanticscholar.org/CorpusID:235484551
22. Bartkiewicz K, Gneiting C, Černoch A, Jiráková K, Lemr K, Nori F (2020) Experimental kernel-based quantum machine learning in finite feature space. Sci Rep 10:68911. https://doi.org/10.1038/s41598-020-68911-5
23. Liu H-L, Yu C-H, Wu Y-S, Pan S-J, Qin S-J, Gao F, Wen Q-Y (2019) Quantum algorithm for logistic regression https://arxiv.org/abs/1906.03834
24. Amin MH, Andriyash E, Rolfe J, Kulchytskyy B, Melko R (2018) Quantum boltzmann machine. Phys Rev X 8:021050. https://doi.org/10.1103/PhysRevX.8.021050
25. Cong I, Choi S, Lukin MD (2019) Quantum convolutional neural networks. Nat Phys 15:1273–1278. https://doi.org/10.1038/s41567-019-0648-8
26. Zhou N-R, Zhang T-F, Xie X-W, Wu J-Y (2023) Hybrid quantum-classical generative adversarial networks for image generation via learning discrete distribution. Image Commun. https://doi.org/10.1016/j.image.2022.116891
27. Hammad A, Nojiri MM, Yamazaki M (2025) Quantum similarity learning for anomaly detection. J High Energy Phys 2025:81. https://doi.org/10.1007/JHEP02(2025)081
28. Ghobadi MZ, Afsaneh E (2024) Potential of quantum machine learning for solving the real-world problem of cancer classification. Discov Appl Sci 6:513. https://doi.org/10.1007/s42452-024-06220-6
29. Pomarico D, Monaco A, Nea Amoroso (2025) Emerging generalization advantage of quantum-inspired machine learning in the diagnosis of hepatocellular carcinoma. Discov Appl Sci 7:205. https://doi.org/10.1007/s42452-025-06638-6
30. Tomono T, Natsubori S (2022) Performance of quantum kernel on initial learning process. EPJ Quantum Technol 9:35. https://doi.org/10.1140/epjqt/s40507-022-00157-8
31. Park S, Park D, Rhee J-KK (2023) Variational quantum approximate support vector machine with inference transfer. Sci Rep 13:3288. https://doi.org/10.1038/s41598-023-29495-y
32. Yang J, Awan AJ, Vall-Llosera G (2019) Support Vector Machines on Noisy Intermediate Scale Quantum Computershttps://arxiv.org/abs/1909.11988

33. Innan N, Khan M, Panda BK et al (2023) Enhancing quantum support vector machines through variational kernel training. Quantum Inf Process 22:374. https://doi.org/10.1007/s11128-023-04138-3
34. Date P, Smith W (2024) Quantum discriminator for binary classification. Sci Rep 14:1328. https://doi.org/10.1038/s41598-023-46469-2
35. El-Shafiey MG, Hagag A, El-Dahshan E-SA (2022) A hybrid ga and pso optimized approach for heart-disease prediction based on random forest. Multimedia Tools Appl 81:18155–18179. https://doi.org/10.1007/s11042-022-12425-x
36. Ogundepo EA, Yahya WB (2023) 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
37. Chakraborty M (2024) Rule extraction from convolutional neural networks for heart disease prediction. Biomed Eng Lett 14(4):649–661. https://doi.org/10.1007/s13534-024-00358-3
38. Simen A, Bloot R, Pires OM, Nascimento EGS (2025) Evolutionary-enhanced quantum supervised learning model. J Supercomput 81(2):388. https://doi.org/10.1007/s11227-024-06875-3
39. Cerezo M, Sone A, Volkoff T, Cincio L, Coles PJ (2021) Cost function dependent barren plateaus in shallow parametrized quantum circuits. Nat Commun 12(1):1791. https://doi.org/10.1038/s41467-021-21728-w
40. Cunningham J, Zhuang J (2025) Investigating and mitigating barren plateaus in variational quantum circuits: a survey. Quantum Inf Process 24(2):48. https://doi.org/10.1007/s11128-025-04665-1
41. Lau JWZ, Lim KH, Shrotriya H, Kwek LC (2022) Nisq computing: where are we and where do we go? AAPPS Bull 32(1):27. https://doi.org/10.1007/s43673-022-00058-z
42. Khang A, Rath KC (2024) (eds.): The Quantum Evolution: Application of AI and Robotics in the Future of Quantum Technology, 1st edn. CRC Press. https://doi.org/10.1201/9781032642079
43. Hirvensalo M (2004) Quantum Computing, 2nd edn. Natural Computing Series. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-662-09636-9
44. Hughes C, Isaacson J, Perry A, Sun RF, Turner J (2021) Quantum Gates, pp. 49–57. Springer, Cham. https://doi.org/10.1007/978-3-030-61601-4_6
45. Baaquie BE, Kwek L-C (2023) Quantum Gates and Circuits, pp. 139–153. Springer, Singapore.https://doi.org/10.1007/978-981-19-7517-2_7
46. Benedetti M, Lloyd E, Sack S, Fiorentini M (2019) Parameterized quantum circuits as machine learning models. Quantum Sci Technol 4(4):043001. https://doi.org/10.1088/2058-9565/ab4eb5
47. Suzuki Y, Yano H, Gao Q et al (2020) Analysis and synthesis of feature map for kernel-based quantum classifier. Quantum Mach Intell 2:9. https://doi.org/10.1007/s42484-020-00020-y
48. Havlíček V, Córcoles AD, Temme K, Harrow AW, Kandala A, Chow JM, Gambetta JM (2019) Supervised learning with quantum-enhanced feature spaces. Nature 567(7747):209–212. https://doi.org/10.1038/s41586-019-0980-2
49. Schuld M (2021) Supervised quantum machine learning models are kernel methods. https://arxiv.org/abs/2101.11020
50. Schuld M, Killoran N (2019) Quantum machine learning in feature hilbert spaces. Phys Rev Lett 122:040504. https://doi.org/10.1103/PhysRevLett.122.040504
51. Schölkopf B, Smola AJ.: (2001)arning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond. Adaptive Computation and Machine Learning series, p. 648. The MIT Press, Cambridge, MA. https://doi.org/10.7551/mitpress/4175.001.0001
52. Hosmer DWJ, Lemeshow S, Sturdivant RX (2013) Applied Logistic Regression, 3rd edn., p. 528. Wiley, Hoboken, NJ, USA. https://www.wiley.com/en-us/Applied+Logistic+Regression%2C+3rd+Edition-p-9780470582473
53. Maalouf M, Trafalis TB, Adrianto I (2011) Kernel logistic regression using truncated newton method. Comput Manag Sci 8:415–428. https://doi.org/10.1007/s10287-010-0128-1
54. Ye JC (2022) 3. Linear, Logistic, and Kernel Regression, pp. 45–59. Springer, Singapore. https://doi.org/10.1007/978-981-16-6046-7_3
55. Misra S (2024) Sythetic Dataset. GitHub. https://github.com/shreshthamisra03/SytheticData_2_features
56. Fisher RA (1936) Iris. UCI Machine Learning Repository.https://doi.org/10.24432/C56C76
57. (Heart) S Statlog (Heart). UCI Machine Learning Repository. https://doi.org/10.24432/C57303
58. Nakai K (1996) Ecoli. UCI Machine Learning Repository. https://doi.org/10.24432/C5388M

# Terms and Conditions

Springer Nature journal content, brought to you courtesy of Springer Nature Customer Service Center GmbH ("Springer Nature").

Springer Nature supports a reasonable amount of sharing of research papers by authors, subscribers and authorised users ("Users"), for small-scale personal, non-commercial use provided that all copyright, trade and service marks and other proprietary notices are maintained. By accessing, sharing, receiving or otherwise using the Springer Nature journal content you agree to these terms of use ("Terms"). For these purposes, Springer Nature considers academic use (by researchers and students) to be non-commercial.

These Terms are supplementary and will apply in addition to any applicable website terms and conditions, a relevant site licence or a personal subscription. These Terms will prevail over any conflict or ambiguity with regards to the relevant terms, a site licence or a personal subscription (to the extent of the conflict or ambiguity only). For Creative Commons-licensed articles, the terms of the Creative Commons license used will apply.

We collect and use personal data to provide access to the Springer Nature journal content. We may also use these personal data internally within ResearchGate and Springer Nature and as agreed share it, in an anonymised way, for purposes of tracking, analysis and reporting. We will not otherwise disclose your personal data outside the ResearchGate or the Springer Nature group of companies unless we have your permission as detailed in the Privacy Policy.

While Users may use the Springer Nature journal content for small scale, personal non-commercial use, it is important to note that Users may not:

1. use such content for the purpose of providing other users with access on a regular or large scale basis or as a means to circumvent access control;
2. use such content where to do so would be considered a criminal or statutory offence in any jurisdiction, or gives rise to civil liability, or is otherwise unlawful;
3. falsely or misleadingly imply or suggest endorsement, approval , sponsorship, or association unless explicitly agreed to by Springer Nature in writing;
4. use bots or other automated methods to access the content or redirect messages
5. override any security feature or exclusionary protocol; or
6. share the content in order to create substitute for Springer Nature products or services or a systematic database of Springer Nature journal content.

In line with the restriction against commercial use, Springer Nature does not permit the creation of a product or service that creates revenue, royalties, rent or income from our content or its inclusion as part of a paid for service or for other commercial gain. Springer Nature journal content cannot be used for inter-library loans and librarians may not upload Springer Nature journal content on a large scale into their, or any other, institutional repository.

These terms of use are reviewed regularly and may be amended at any time. Springer Nature is not obligated to publish any information or content on this website and may remove it or features or functionality at our sole discretion, at any time with or without notice. Springer Nature may revoke this licence to you at any time and remove access to any copies of the Springer Nature journal content which have been saved.

To the fullest extent permitted by law, Springer Nature makes no warranties, representations or guarantees to Users, either express or implied with respect to the Springer nature journal content and all parties disclaim and waive any implied warranties or warranties imposed by law, including merchantability or fitness for any particular purpose.

Please note that these rights do not automatically extend to content, data or other material published by Springer Nature that may be licensed from third parties.

If you would like to use or distribute our Springer Nature journal content to a wider audience or on a regular basis or in any other manner not expressly permitted by these Terms, please contact Springer Nature at

onlineservice@springernature.com