scientific reports

OPEN

Robust evaluation of classical and quantum machine learning under noise, imbalance, feature reduction and explainability

Savita Kumari Sheoran[1], Vikesh Yadav[1]✉ & Rakesh Kumar Sheoran[2]

Quantum machine learning (QML) has emerged as a promising paradigm for solving complex classification problems by leveraging the computational advantages of quantum systems. While most traditional machine learning models focus on clean, balanced datasets, real-world data is often noisy, imbalanced and high-dimensional, posing significant challenges for scalability and generalisation. This paper conducts an extensive experimental evaluation of five supervised classifiers- Decision Tree, K nearest neighbour, Random Forest, linear regression and support vector machines in comparison with Quantum machine learning classifiers- quantum Support vector machine, quantum k- nearest neighbor and variational quantum classifier—across five diverse datasets, including iris, wine quality, Breast cancer, UCI human activity recognition, and Pima diabetes. To simulate real-world challenges, we introduce class imbalance using SMOTE and ADASYN Sampling, inject Gaussian noise into the features, and assess the impact of dimensionality reduction through ANOVA-based feature selection. Additionally, we utilise explainable AI tools, such as SHAP and LIME, to interpret model decisions. Our results demonstrate that Logistic Regression consistently performs well under various complexities, while Quantum Support Vector Machines show resilience to feature noise and class imbalance. The study also highlights the current capabilities and limitations of QML models, offering valuable insights into building generalisable and interpretable ML systems for deployment in complex environments. These insights are crucial for building robust, interpretable, and generalisable ML models for practical deployment.

Keywords Machine learning, Classification, Linear regression, Random forest, Support vector machine, K-nearest neighbor, Decision tree, Quantum support vector machine

Machine Learning (ML) is employed in various computational domains to enhance performance and accuracy. However, the morphology of datasets used to learn the machine offers some obstacles in this task. Machine learning datasets typically consist of many tuples and a limited number of characteristics. Microarray technology, capable of exhibiting particular distinctions from conventional machine learning datasets, is deployed to overcome the morphological issues. To handle complex and vast varieties of datasets, various types of machine learning algorithms, viz. supervised, unsupervised or semi-supervised, are used. Supervised learning algorithms are the preferred machine learning approaches for classification and regression-related tasks. The Support Vector Machine (SVM), Random Forest (RF), Neural Network (NN), Linear Regression, Decision Tree, & K-Nearest Neighbor are some extensively used supervised learning methods found in the literature[1].

Machine Learning entails the development of a prediction algorithm based on past experiences, which necessitates obtaining relevant data in the specific field. Subsequently, the prediction network self-organizes based on the margin of error[2] . In the present day, extracting important information from raw data for efficient decision-making is crucial in business, scientific, medicine, science and engineering applications. Modern intelligence technologies employ data analysis to examine and transform information into knowledge. Data Mining (DM) and Machine Learning (ML) are crucial in accurately extracting information from large datasets. Several machine-learning approaches are available for prediction, including classification, clustering, decision making and regression, as depicted in Fig. 1 below[3].

[1]Dept. of Computer Science & Engineering, Indira Gandhi University Meerpur, Rewari, Haryana, India. [2]SIASTE, Gurugram, Haryana, India. ✉email: vikeshyadav1994@gmail.com

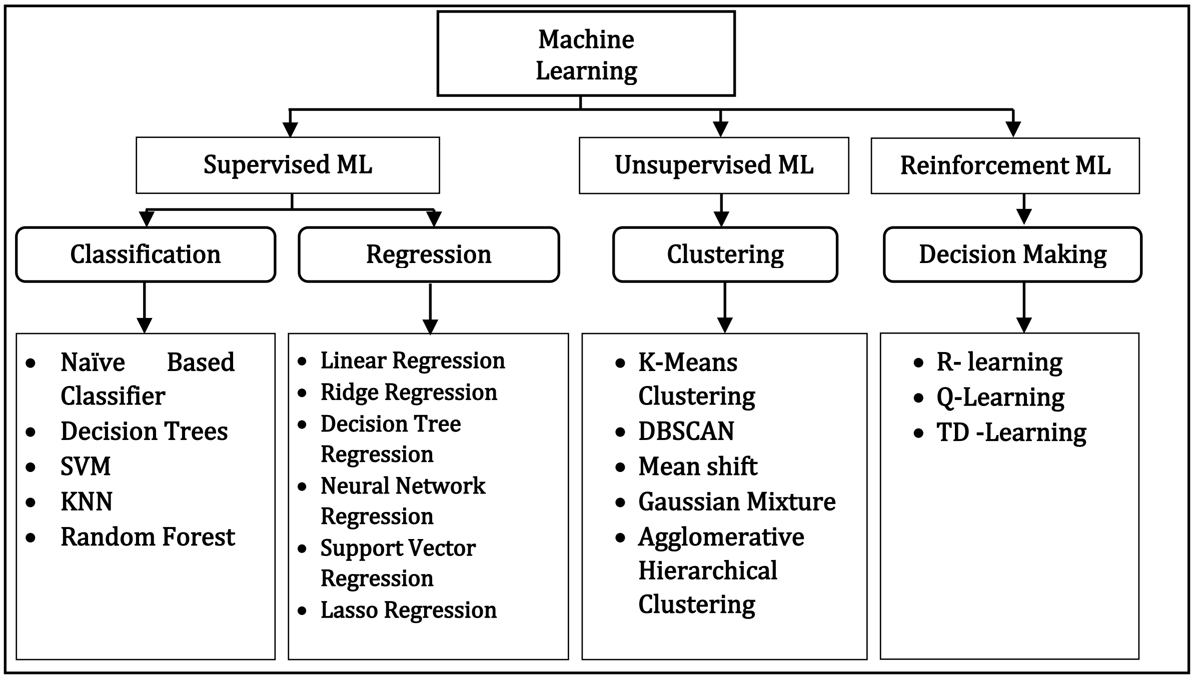


Fig. 1. Machine learning techniques.

Machine Learning enables systems to acquire knowledge from insight into datasets and improve their performance through natural learning processes without requiring explicit programming. Such algorithms are valuable in domains where it is impractical to implement explicitly written algorithms that can achieve high-speed performance[4].

Classical Machine learning algorithms face limitations in scalability and computational efficiency, especially when the dataset size grows, resulting in growing complexity. And this is where we introduce Quantum machine learning, which leverages the principles of Quantum computing, such as superposition, entanglement, and quantum parallelism, to process information in fundamentally new ways. i.e., a Quantum Support Vector Machine (QSVM) can utilise quantum-enhanced feature maps and kernel estimation to classify data, potentially achieving speedups over classical SVMs, especially in high-dimensional feature spaces. Similarly, Quantum Decision Trees and Quantum Neural Networks (QNNs) explore non-classical representations and optimisations of classical models[3].

The objective of quantum machine learning is to surpass the classical limits of problems such as pattern recognition, probabilistic mapping and optimisation. Although QML algorithms are currently in the early stages due to the limited availability of qubits and restrictions on quantum hardware, initial results show the potential of QML in the real world. Figure 2 illustrates the evolution of Quantum machine learning from classical machine learning.

The goal of the study is to assess the robustness and interpretability of both classical and quantum classifiers under realistic data challenges. The output of this research will be helpful in selecting a classifier that effectively evaluates previous data and makes accurate predictions for future decisions. This study makes the following key contributions:

- Conducts a comprehensive empirical evaluation of five widely used supervised machine learning classifiers-Decision Tree, K-Nearest Neighbor, Random Forest, Logistic Regression, Support Vector Machine and Quantum Machine Learning classifiers- quantum Support vector machine(QSVM), quantum k-nearest neighbor (QKNN) and variational quantum classifier (VQC) across five publicly available datasets, reflecting varied real-world conditions.
- Introduces class imbalance using both SMOTE and ADASYN and injects Gaussian noise into input features to simulate noisy high-dimensional environments commonly encountered in practical applications.
- Applies ANOVA-based univariate feature selection to evaluate the effect of dimensionality reduction on model performance and to prepare inputs for quantum circuits.
- Analyses the performance of all classifiers under varying conditions of data noise, imbalance, and feature sparsity using standard evaluation metrics.
- Employs explainable AI methods, including SHAP and LIME, to interpret model decisions, and quantum kernel distribution, enhancing model transparency and trustworthiness, with extensions to quantum classifier interpretability.
- Identifies logistic regression as a consistently high-performing model under noise and imbalance, while QSVM demonstrates resilience to noisy features; provides insights on the current capabilities of QML classifiers.

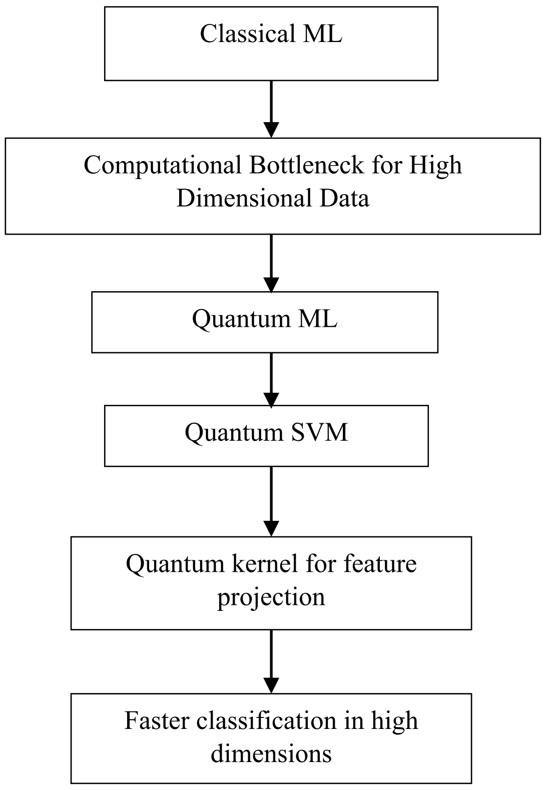

Classical ML
Computational Bottleneck for High Dimensional Data
Quantum ML
Quantum SVM
Quantum kernel for feature projection
Faster classification in high dimensions

Fig. 2. Transition from classical ML to quantum ML.

Section "Introduction" explains the concepts of supervised machine learning and Quantum machine learning. Sect. Related work presents an analysis of existing studies in the fields of machine learning and quantum computing. Section "Methodology" describes the methodology, followed by the results section, and Sect.conclusion concludes the paper.

Related work

Several researchers have conducted extensive studies on data analysis through Machine Learning (ML) and Quantum computing (QC) methodologies. Multiple studies have indicated the importance of these strategies in predicting future outcomes, particularly in the realm of classification problems. In these investigations, the authors utilised various methods to address specific issues and achieved high levels of accuracy in categorisation. For instance, these strategies are employed in the healthcare sector to predict diseases. Ch Anwar ul Hassan et al. evaluated the effectiveness of machine learning classifiers. Various ML classifiers, including LR, Naive Bayes, KNN and DT, were used to compare the accuracy, precision, and F-measure of the two datasets. The experimental results demonstrate that Random Forests outperform alternative classifiers. The model achieves an accuracy of 83% in heart data sets and 85% accuracy in predicting hepatitis illness[5]. Arslan Javaid et al. Proposed an innovative classification methodology and divided skin lesions into benign or malignant categories, employing image processing techniques with machine learning algorithms. Their work presented an innovative method for enhancing the contrast of thermoscope images and the OTSU thresholding technique utilised for picture segmentation. Subsequently, the feature vector undergoes standardisation and scaling. Before classification, a unique feature selection technique based on the wrapper method is proposed. The Random Forest method is the most successful and accurate classification algorithm for achieving maximum accuracy[6]. Vaishnavi Nath et al. created and implemented an innovative fraud detection technique to analyse streaming transaction data and study consumers' past transaction information to identify their behavioural patterns. Cardholders are classified into separate categories based on their transaction amounts. Next, employing the sliding window approach, the exchanges performed by cards from several groups are combined to take out the respective behavioural patterns of each group. Subsequently, distinct classifiers are trained on each group individually[7]. Rabia Karakaya et al. examined the operational rationale of the method for recognising handwritten digits and evaluated the effectiveness of several methods on the identical database. A report was presented by doing a comparative analysis of the accuracy[8]. Piyush Vyas et al. proposed a hybrid approach, combining techniques based on lexicon for examining and labelling tweet sentiment with supervised machine learning approaches for classifying tweets and assessed the hybrid framework using numerous metrics for evaluation. The findings suggest that a significant proportion of the attitudes expressed are positive (38.5%) or neutral (34.7%)[9]. A summary of research in the field of quantum and classical machine learning is presented in Table 1.

Ref	Dataset	Techniques	Result	XAI	Noise	Feature selection	Year
[10]	RoEduNet-SIMARGL2021, CICIDS-2017	RF, ADA, DNN, SVM, KNN, MLP, LIGHT GBM	99%	Yes	No	Yes	2024
[11]	Audio sensor data	SVM, RF, LR, GNB, EGB	80.28%	Yes	No	Yes	2024
[12]	Case Western Reserve University (CWRU) bearing dataset	Convolutional Long Short-Term Memory (CLSTM)	n/a	Yes	No	Yes	2024
[13]	ShipsEar dataset and simulated submarine data	RF, ADA boost, GBDT, X boost	94.5%,76%,95%,96.7%	No	No	Yes	2023
[14]	Phase classification dataset	QSVM, VQC	97.73,96.49	Yes	No	Yes	2025
[15]	MNIST FMNIST, KMNIST, and CIFAR10	Quantum autoencoder + VQC	65%	Yes	No	No	2025
[16]	MNIST, Ionosphere, waveform, Madelon, synth_10, synth_100	QUBO	90, 78,87	No	No	Yes	2025
[17]	Wisconsin breast cancer data, kaggles's Club data	QUBO	69,62, 72.06	No	No	Yes	2023

Table 1. Summary of research in the field of classical and quantum ML.

Literature review gap

An in-depth analysis of the recent literature from 2023 to 2025 on machine learning and quantum machine learning applications across diverse datasets (e.g., RoEduNet-SIMARGL2021, CICIDS-2017, CWRU bearing, ShipsEar, MNIST, and others) reveals significant progress in terms of classification accuracy and the integration of explainable AI (XAI) and feature selection techniques. However, none of the reviewed studies have simultaneously addressed all three critical components: XAI integration, feature selection, and the inclusion of realistic quantum or classical noise models. This gap indicates a pressing need for a unified framework that can incorporate:

- Explainable AI (XAI) for model interpretability,
- Feature selection to reduce dimensionality and enhance performance, and
- Noise models to simulate real-world deployment environments, particularly in quantum machine learning.

Addressing this triad holistically could significantly enhance the reliability, transparency, and deployment ability of both classical and quantum machine learning models.

Methodology

The goal of this section is to assess the performance and interpretability of five widely used classical ML algorithms -support vector machine, random forest, logistic regression, decision Tree and k nearest neighbor, along with three quantum ML algorithms- Quantum SVM, quantum kNN and variational Quantum Classifier, under various levels of complexity, i.e. noise handling, resampled data. Various steps of the methodology is illustrated in Fig. 3.

Key Python modules and libraries used include:

- Scikit-learn: For data preprocessing, feature selection, classical model training, and performance evaluation.
- Imbalanced-learn: Specifically, SMOTE and ADASYN were used to address class imbalance in the training data.
- PennyLane: For constructing and simulating quantum circuits and implementing the Quantum ML Models.
- SHAP and LIME: For model explainability and visualisation, aiding in the interpretation of feature contributions to model decisions.
- Matplotlib and Seaborn: For plotting ROC curves and explanation visualisations.
- NumPy and Pandas: For efficient numerical computation and data manipulation.

Each step is described in detail below.

Here is the pseudo-code for the methodology, for both the quantum and classical models:

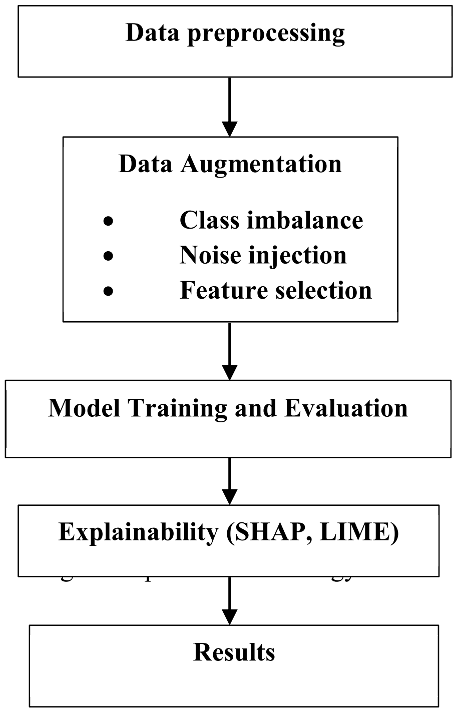


Fig. 3. Proposed Methodology.

```
START
DEFINE datasets = [Iris, Wine Quality, Breast Cancer, HAR, Pima Diabetes, Credit Card Fraud]

FOR each dataset IN datasets:
  // Data Loading and Preprocessing
  LOAD features X and labels y
  SPLIT X, y into X_train, X_test (80/20 stratified split)
  STANDARDIZE X_train and X_test
  // Class Imbalance Handling
  APPLY SMOTE → X_train_balanced, y_train_balanced
  APPLY ADASYN

  // Feature Noise Injection

  FUNCTION add_gaussian_noise(X, factor):
    RETURN X + factor * Gaussian_noise
  X_train_noisy = add_gaussian_noise(X_train_balanced, factor=0.1)
  X_test_noisy = add_gaussian_noise(X_test, factor=0.1)
  //Feature Selection (ANOVA)
  SELECT top-k features using ANOVA F-test
  TRANSFORM X_train_balanced, X_test → X_train_selected, X_test_selected
  //Classical Classifier Evaluation
  DEFINE classifiers = {
    "Decision Tree": DecisionTreeClassifier(),
    "KNN": KNeighborsClassifier(),
    "Random Forest": RandomForestClassifier(),
    "SVM": SVC(),
    "Logistic Regression": LogisticRegression()
  }

  FOR each classifier_name, model IN classifiers:
    FOR scenario IN ["Clean", "Balanced", "Noisy", "Selected Features"]:
      SELECT corresponding X_train and X_test
      TRAIN model on X_train
      PREDICT on X_test
      CALCULATE metrics: accuracy, precision, recall, F1
      STORE metrics in results[classifier_name][scenario]
  //Quantum Model Evaluation (QSVM/QNN/VQC)
  REDUCE input features via PCA/ANOVA (k ≤ 8)
  MAP features to quantum space using Angleencoding
  COMPUTE quantum kernel (QSVM) OR DEFINE variational circuit (QNN/VQC)
  TRAIN quantum model on training data
  PREDICT on test data
  CALCULATE metrics: accuracy, precision, recall, F1
  STORE metrics in results["Quantum Models"]
  // Explainability
  //Classical Models
  SELECT best performing classical model
  APPLY SHAP (TreeExplainer / KernelExplainer)
  PLOT global feature importance
  APPLY LIME on random test instances
  DISPLAY local explanations
//Quantum Models
  ANALYZE feature contributions via quantum kernel
  VISUALIZE kernel heatmaps or fidelity-based importance
  //Result Storage
  CONVERT results to tabular format
  SAVE results for this dataset

END FOR
```

Algorithm. Data preparation, noise handling, feature selection, and model evaluation

Datasets

The different datasets have been selected to assess the performance of QML and ML classifiers. These datasets have been retrieved from the UCI and Kaggle repositories, as tabulated in Table 2. The term is used in the dataset to refer to several instances, features, classes, and their corresponding null values, if present.

Dataset Name	Features	Features used for classical ML Models	Features used for Quantum ML Models	Instances	Class Distribution	Purpose
Iris	4	4	4	150	Balanced (3 classes)	Classic classification dataset
Wine Quality	13	13	6	178	Imbalanced	Chemical analysis of wine
Breast Cancer Wisconsin	30	30	6	569	Imbalanced	Medical diagnosis
Human Activity Recognition	561	561	6	10,299	Balanced (6 activities)	Real-world sensor data
Pima Indian Diabetes	8	8	6	768	Imbalanced	Medical data

Table 2. Structure of the Datasets.

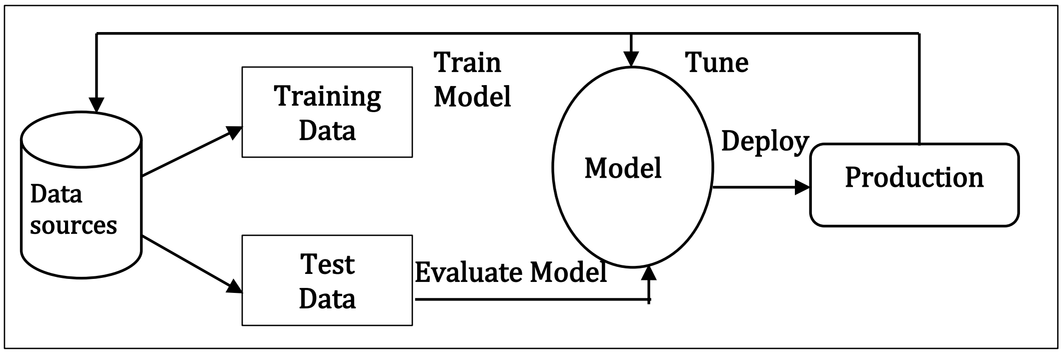


Fig. 4. Workflow of Su learning.

Data preprocessing

Data preprocessing ensures the datasets are ready for model training and testing. The steps followed are:

Missing values
Missing values (if any) are imputed using the median (for numerical features) and mode (for categorical features).

Feature scaling
Standardization (z-score normalisation) is applied to all continuous variables to ensure uniformity.

Class imbalance simulation
SMOTE (Synthetic Minority Over-sampling Technique): Used for over-sampling the minority class in imbalanced datasets like Wine Quality, Breast Cancer, and Credit Card Fraud.

ADASYN (adaptive synthetic sampling): Applied over-sampling to the minority class in specific experiments, especially for highly imbalanced datasets. It generates synthetic samples adaptively, focusing more on minority class instances that are harder to learn, thereby improving classifiers' performance.

Noise injection
Gaussian noise is added to the features to simulate sensor or data measurement errors. The noise level is controlled (e.g., adding 1% noise by randomly sampling from a Gaussian distribution).

$$X_{noisy} = X + N\left(0, \sigma^2\right) \tag{1}$$

where $N\left(0, \sigma^2\right)$ Represents Gaussian noise with mean zero and standard deviation $\boldsymbol{\sigma}$.

Feature selection
SelectKBest: A univariate feature selection method based on ANOVA F-test scores is used to reduce the feature set and evaluate model performance under reduced dimensionality.

Dimensionality Reduction: The top k features (using SelectKBest) are selected to simulate a scenario with reduced information.

Supervised and quantum machine learning classifiers

Supervised learning in machine learning involves acquiring a function that establishes a relationship between input and output data, which can be achieved by utilising provided examples of input–output pairs. The procedure entails deriving a mathematical function based on training data labelled by assigning certain categories or classes and comprising a collection of training instances. The input dataset is partitioned into many training and testing datasets, as illustrated in Fig. 4. The training dataset includes a target variable that necessitates forecasting or

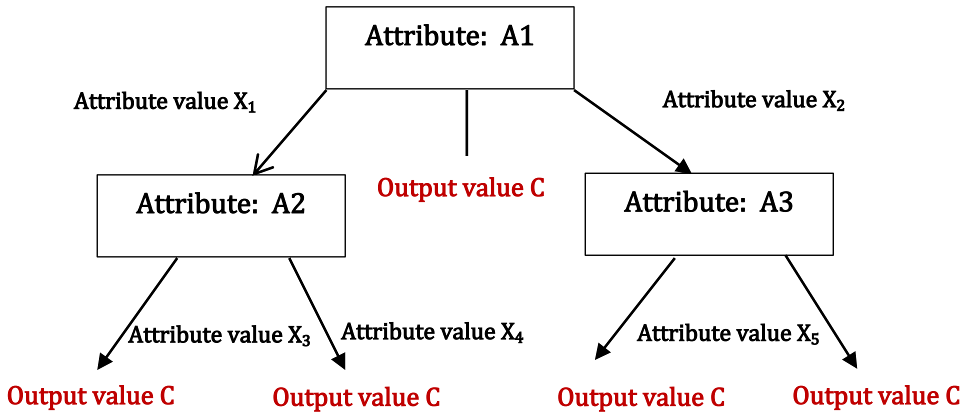


Fig. 5. Instance of decision tree.

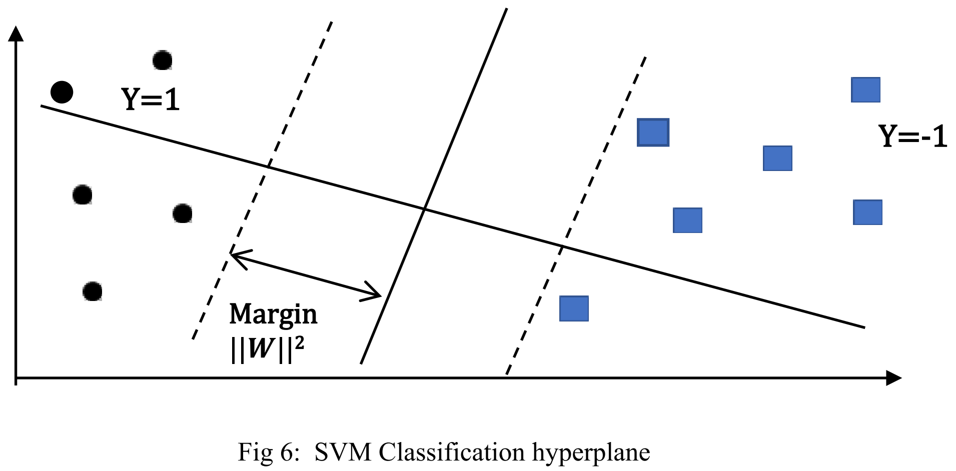


Fig. 6. SVM Classification hyperplane.

categorisation. Algorithms extract patterns from the training dataset and employ them to make predictions or categorise the test dataset[18].

Currently, Quantum ML has emerged as a promising advancement of classical ML by leveraging the principles of Quantum Computing to enhance the capacity of learning models. Quantum Models follow the same learning from labelled data but they encode the classical data to quantum data using feature maps and process it using quantum circuits. Figure 4 depicts the pipeline of machine learning.

Decision tree (DT)

Constructing a classifier involves utilising a set of internal and leaf nodes. The internal nodes represent decision criteria, while predictions are represented by the leaf nodes. Let $'N'$ Be the quantity of physiological characteristics and $'M'$ Represent the number of diagnosis predictions. Let $'X'$ Be a set of physiological characteristic vectors denoted $as\,\{X_1, X_2.........X_N\}$. Let $'W'$ Be the set of $'N'$ thresholds, denoted as $\{W_1, W_2....................W_N$. Consider $'C'$ as a set of diagnosis predictions, denoted as $as\,\{C_1, C_2.........C_N\}$.. This process commences from the root node and proceeds towards the last node. Consequently, the classification effect depends on the choices from the starting node to the finishing node. The classification of a classifier based on decision trees can be understood as a collection of IF–THEN statements by systematically visiting each decision route. Hence, it is probable to transform a DT classifier in any rule base consisting of the collections of IF–THEN rules that are formed from the DT classifier[19]. The attribute that each node tests is labelled, and the values that correspond to those labels are labelled on its branches, as illustrated in Fig. 5.

Support vector machine

SVM, a supervised classification method, begins with a pre-existing training set. During the training process, a Support Vector Machine (SVM) acquires knowledge regarding the correlation between each data point and its related label in the given training dataset[8]. It is specifically designed for binary classification of new testing vectors and refers to a type of issue characterised by having more equations than unknowns, which is known as an over-determined system of equations as represented in Fig. 6. The algorithm produces a hyperplane defined by the equation $\rightarrow_{w}\ .\ \rightarrow_{x}+\mathrm{b}=0$. This hyperplane ensures that to obtain a training point $\rightarrow_{xn}$ in the positive class, $\rightarrow_{w}\ .\ \rightarrow_{x}+\mathrm{b}\geq 1$, and to obtain a training point $\rightarrow_{\mathrm{xn}}$ in the negative class, $\rightarrow_{w}\ .\ \rightarrow_{x}+\mathrm{b}\leq—1$. The algorithm strives to optimise the distance between the two classes throughout the training process. This is logical since we wish to create a clear separation between the two classes in order to obtain a more accurate classification result for new data samples, such as $\rightarrow_{\mathrm{xo}}$. Mathematically, Support Vector Machines (SVM) find any hyperplane that maximises the distance between two parallel hyperplanes, subject to the restriction that the product of the predicted label ($\rightarrow_{\mathrm{yi}}$) and the linear combination of the weights ($\rightarrow_{\mathrm{w}}$) and the input vector ($\rightarrow_{\mathrm{xo}}$)[20]. The diagram in Fig. 6 below represents the categorization of classes using a hyperplane.

K–Nearest Neighbor (KNN)
An enduring approach in class reasoning. The decision-making concept is based on a straightforward principle: the sample that needs to be evaluated is classified in the same category as the closest matching sample. The outcome of the Nearest Neighbor Rule is definitively established for all occurrences to be evaluated, assuming that the distance metric and training set remain constant. In set E, for every sample instance, if y is the closest neighboring instance to x, then the group of y is determined by the nearest neighbor rule. Assume X is a sample from an unidentified category. The decision process is envisaged in equation (2):

$$g_j\left(X\right)=\min g_i X)i=1,2,...,\tag{2}$$

Subsequently, the outcome of the choice is $X\varepsilon Wi$.

In this context, the nearest neighbour rule is presented with a focus on two key aspects: convergence and generalisation error. The nearest neighbour for a given point x, derived from two training sets with different samples, varies. Given that the classification outcome is contingent upon the category label of the closest neighbouring data point, $P\left(e|x,x'\right)$, is so obtained. x and $x\prime$ factors determine the conditional error rate[21].

Random forest (RF)
This model is a collective learning technique. The algorithm constructs a series of decision trees throughout the training process and the mode determines the class that has the highest frequency among the trees for classification tasks, while it calculates the mean prediction for regression tasks. The node splitting criterion was configured to demand a minimum of two samples, and each leaf node must include at least one sample. The procedural instructions of the Random Forest algorithm:

- Generate a bootstrap sample using the given data.
- In order to create each bootstrap sample, a regression tree must be constructed with certain modifications: randomly select a subset of the predictors at each node and determine the optimal split among the variables.
- Calculate the most recent data by summing up the forecasts of the n_t trees (taking the average for regression).

Built on a random selection of observations, whether with or without replacement, approximately 36.8% are not utilized for any one tree. This means that these observations are considered "out of the bag (OOB)" for that specific tree. The predictive accuracy of a random forest can be assessed using the out-of-bag (OOB) data.

$$OOB-MSE=\frac{1}{n}\sum_{i=1}^{n}\left(y_i-\widehat{y_i}OOB\right)^2\tag{3}$$

where the average forecast for the i_{th} opinion from total trees the observation that is out-of-bag (OOB) is represented by $\widehat{y_i}OOB$.

Linear regression (LR)
Regression is a method employed to analyze the association between two variables. The analysis is commonly employed for the purpose of forecasting and prediction, and it shares significant similarities with the field of machine learning. Crucially, this method demonstrates the relationships between a dependent variable and a predetermined set of other variables in a dataset. The equation is represented as $y=\beta 0+\beta 1x+\varepsilon$. Simple regression separates the impact of independent factors from the interplay of dependent variables.

Quantum support vector machine
A Quantum Support Vector Machine (QSVM) is a quantum-enhanced version of the classical Support Vector Machine (SVM) algorithm, designed to perform classification tasks. QSVM leverages quantum computing, particularly quantum kernel methods, to potentially outperform classical algorithms in certain scenarios, especially when dealing with high-dimensional data or complex patterns that are difficult to capture classically.

Quantum SVM replaces the classical kernel function with a quantum kernel, which is computed by a quantum computer. This quantum kernel is based on the inner product of quantum states, which allows encoding classical data into quantum feature space.

Quantum kernel estimation
The key idea is to map a classical data point xxx to a quantum state $|\phi\left(x\right)\rangle|$ and then compute a kernel matrix $K\left(x,x'\right)=|\langle\phi\left(x\right)|\ \ \phi\left(x'\right)\rangle|^2$ This is usually done by:

Encoding data into quantum circuits using a feature map $U\phi\left(x\right)$.

Measuring the fidelity (overlap) between quantum states

This allows capturing nonlinear patterns efficiently, potentially with exponential speedups.

Workflow of QSVM
Feature mapping: Encode classical data xxx into a quantum state using a parameterized unitary $U\phi\left(x\right)$.

Kernel evaluation: Compute $K\left(x,x'\right)=\left|\left\langle \phi\left(x\right)\right|\ \phi\left(x'\right)\rangle\right|^2$.

Classical SVM Solver: Use a classical algorithm (e.g., LIBSVM) to find the separating hyperplane using the quantum kernel matrix

Prediction: New data points are mapped and evaluated using the quantum kernel.

Quantum k nearest neighbor
The Quantum kNN algorithm adapts the classical k nearest neighbor classifier to a quantum computing paradigm by representing classical vectors as quantum states and using quantum subroutine to estimate inter sample similarities in superposition. QkNN aims to accelerate distance or similarity estimation and nearest neighbor search under theoretical models that provide efficient quantum access to data[22].

QkNN implementations usually estimate $s\left(x_q,x_i\right)$ using one of the:

- Fidelity / inner product: $\left|\left\langle x_q\right|\ x_i\rangle\right|^2$, estimated via a SWAP test or Hadamard test.
- Euclidean distance via inner product identity:
- $\parallel x_q-x_i\parallel^2=\parallel x_q\parallel^2+\parallel x_i\parallel^2-2\left\langle x_q,x_i\right\rangle$ where the inner product is obtained from amplitude overlaps.
- Hamming or other discrete distances when features are binarised; specialised circuits exist for parallel Hamming distance estimation.

Variational quantum classifier
The Quantum Variational Classifier (VQC) is a hybrid quantum–classical supervised learning model that leverages parameterised quantum circuits (PQCs) optimised via classical algorithms. It belongs to the class of Variational Quantum Algorithms (VQAs), specifically designed for Noisy Intermediate-Scale Quantum (NISQ) devices.VQC combine quantum feature encoding and trainable quantum layers to learn complex, nonlinear decision boundaries, analogous to a neural network but implemented on qubits. Their advantage lies in exploiting quantum entanglement and superposition to represent data in exponentially large Hilbert spaces, offering potential expressivity beyond classical models[23] . If given a dataset D, then

$$D=\left\{\left(x_i,y_i\right)\right\}_{i=1}^N,x_i\in\mathbb{R}^d,y_i\in\left\{-1,+1\right\}, \tag{4}$$

the goal is to find a variational quantum circuit $U\left(\theta\right)$ parameterised by a vector of tunable parameters θ such that the measurement expectation value corresponds to the predicted class label.

This process these steps are included :

- Encoding: map classical data x_i into a quantum state $\left|x_i\right\rangle=U_\phi\left(x_i\right)\left|0\right\rangle^{\otimes n}$, where U_ϕ is a data-dependent unitary transformation.
- Parameterised evolution: Apply a trainable unitary $U\left(\theta\right)$ to form $\left|\psi\left(x_i,\theta\right)\right\rangle=U\left(\theta\right)U_\phi\left(x_i\right)\left|0\right\rangle^{\otimes n}$

Measurement: Measure an observable (e.g., Pauli-Z) to obtain the expectation value:

$$f_\theta\left(x_i\right)=\left\langle\psi\left(x_i,\theta\right)\right|M\left|\psi\left(x_i,\theta\right)\right\rangle,$$

where M is the measurement operator. The sign of $f_\theta(x_i)$ determines the class label.

Optimisation: Minimise a cost function such as binary cross-entropy or hinge loss[24]:

$$L\left(\theta\right)=\frac{1}{N}\sum_i\ell\left(f_\theta\left(x_i\right),y_i\right) \tag{5}$$

Evaluation parameters
Within the domain of data mining, several evaluation measures, including accuracy, F-measure, precision, and recall, are commonly employed to assess the system's performance. In this paper, we will consider these measures to evaluate performance. TPe is an abbreviation for correctly identified positive, FPe is a false positive, TNe is a true negative, and FNe is a false negative.

The accuracy rate is the ratio of instances with correct classification to the total cases tried, as stated in Eq. (1). The equation can be formulated in the following manner:

$$\text{Accuracy}=\frac{(TPo)+(TNe)}{(TPo+(FPo)\ +(TNe)+(FNe)} \tag{6}$$

The **precision** of a model is defined as the proportion of the truly positive instances that are correctly forecasted to all positive forecasts generated by the model.

$$\text{Precision}=\frac{(TrP)}{(TrP)+(FlP)} \tag{7}$$

The **recall** calculates the percentage of positive scenarios correctly predicted to the total number of positive scenarios, with the false negative scenarios, and is occasionally called the true positive rate.

$$\text{Recall} = \frac{(TrP)}{(TrP) + (FlN)} \tag{8}$$

The **F1 score** is a quantitative measure that assesses the relationship between precision and recall. The calculation involves taking the harmonious average of both. It is a valuable metric for achieving a compromise between high precision and strong recall. It effectively penalises extreme negative values of either component.

Accuracy quantifies the overall correctness or precision of a measurement or calculation. In the equation $Pr1$ stands for precision $Re1$ stands for recall

$$\text{F1 score} = \frac{2 \times \ (\text{Pr1} \times \ \text{Re1})}{(\text{Pr1} + \ \text{Re1})} \tag{7}$$

ROC curve: The Receiver Operating Characteristic (ROC) curve is a standard tool for evaluating binary classifiers, especially in imbalanced medical datasets. It illustrates the trade-off between a True Positive Rate (Sensitivity) and a False Positive Rate (1—Specificity).

Confusion matrix: It is a tabular representation of showing the counts of True positives, true negatives, false positives and false negatives.

Explainability: SHAP, LIME and Quantum kernel distribution

To enhance model transparency, we use two popular **Explainable AI (XAI)** techniques: **SHAP** and **LIME**. These techniques provide insights into which features are driving the model's predictions.

SHAP

We use SHAP (SHapley Additive exPlanations) to evaluate feature importance globally. SHAP values indicate the impact of each feature on the prediction for each instance.

LIME

For local interpretability, we use LIME to explain individual predictions by approximating the model with simpler interpretable models around each prediction.

Quantum kernel distribution

To analyse quantum-enhanced models like QSVM and QKNN, we examine the distribution of quantum kernel values. The quantum kernel distribution provides insights into how well different classes are separable in the high-dimensional feature space, highlighting regions where the quantum embedding improves classification performance.

Hardware availability and simulation justification

All quantum modes in the study are implemented and executed using the pennylane framework on Google Colab, leveraging its high-performance cloud runtime for reproducible quantum simulation. This design choice was guided by the limited public availability of current NISQ devices, where hardware execution remains constrained by qubit decoherence and hardware noise and is not publicly available or available at a very high cost. Running on Google Colab ensures platform independence and reproducibility across multiple runs. The implemented circuits are intentionally designed with shallow depth and fewer qubit requirements. Consequently, the reported results represent a reproducible and hardware-ready baseline for future near-term deployment.

Experimental results and analysis

Supervised and Quantum Machine Learning classification acquires knowledge from data that is not evaluated based on its performance; eighty per cent of the dataset was designated for training. while the remaining 20% of the dataset is designated as the testing set. The model's performance is assessed on unseen testing sets, where the model learns from the data without adjusting and optimizing the parameters.

Parameter tuning

Table 3 summarizes key experimental settings, covering dataset handling, preprocessing, model configurations, and evaluation strategies. Each component-from feature selection and resampling to classical and quantum classifiers-was tuned or applied with specific parameters to ensure fair, reproducible, and interpretable results.

Performance overview of classical ML models

To evaluate the performance and robustness of classical machine learning models under varying data conditions, we conducted experiments on five benchmark datasets: Breast Cancer Wisconsin, Iris, Wine Quality, Pima, HAR human activity Dataset. Each dataset was subjected to four experimental variants: (1) resampled data using SMOTE (2) ADASYN (3) noisy data with Gaussian noise, and (4) selected features using ANOVA F-test. Performance was assessed via accuracy, precision, recall, F1-score, confusion matrix and Area Under the ROC Curve (AUC). The results are reported for both original and resampled datasets using SMOTE to address class imbalance. Tables 4, 5, 6, 7 and 8 summarize the classification performance across all datasets and models. For clarity and reproducibility, each table is grouped by ML Models and includes all four data preprocessing variants.

Sr. No	Category	Parameter	Value/setting	Description/purpose
1	Dataset splitting	test_size	0.2	80% training, 20% testing split
		random_state	42	Seed for reproducibility across runs
		stratify	y (class labels)	Ensures class proportion balance during split
2	Preprocessing	StandardScaler	Mean = 0, Std = 1	Standardizes each feature to zero mean, unit variance
		noise_factor	0.01	Standard deviation for Gaussian noise injection
		add_noise()	$(X' = X + N\left(0, \sigma^2\right)$	Simulates real-world perturbations (used for "Noise" runs)
3	Imbalance handling	SMOTE	random_state = 42	Synthetic Minority Oversampling Technique for balancing minority classes
		ADASYN	Supported	Adaptive synthetic sampling (disabled for HAR to avoid instability)
4	Feature selection	SelectKBest	score_func = f_classif, k = 6	ANOVA F-test selects top 6 informative features
		PCA (for QSVM)	n_components = 4	Dimensionality reduction to 4D latent space for quantum embedding
5	Classical ML classifiers	Support vector machine (SVM)	$kernel =' rbf', C \in \{0.1, 1, 10\}, gamma \in \left\{'scale','auto'\right\}$	Baseline non-linear classifier; tuned via fivefold GridSearchCV
		K-Nearest Neighbors (KNN)	$n\_neighbors \in \{3, 5, 7, 9\}$, $weights \in \left\{'uniform','distance'\right\}$	Distance-based local voting classifier
		Random forest (RF)	$n\_estimators \in \{50, 100, 200\}$, $max\_depth \in \{None, 5, 10\}$	Ensemble of decision trees with bagging
		Decision tree (DT)	$criterion \in \left\{'gini','entropy'\right\}$, $max\_depth \in \{None, 5, 10\}$	Single interpretable tree baseline
		Logistic regression (LR)	$C \in \{0.1, 1, 10\}$, $solver \in \left\{'lbfgs','liblinear'\right\}$, $max\_iter = 2000$	Linear baseline classifier with L2-regularization
6	Quantum classifier	Quantum device	qml.device("default.mixed", wires = 8)	Simulated 8-qubit Pennylane backend for hybrid experiments
		Feature embedding	qml.templates.AngleEmbedding(x)	Encodes normalised features as qubit rotation angles
		Quantum kernel distribution	Histogram of $(K\,(x_i, x_j))$	Shows the spread of kernel values, used to visualise class separability
		Quantum kernel	Overlap kernel $(K\,(x_i, x_j))$	$\psi\ (x\_i)\rangle$
		QSVM	SVC(kernel = 'precomputed')	Uses a quantum kernel matrix instead of a classical kernel
		QKNN	KNeighborsClassifier(n_neighbors = 3)	Distance metric derived from quantum kernel space
		VQC (Variational Quantum Classifier)	AngleEmbedding + BasicEntanglerLayers, 2 layers, step size 0.2, 10 epochs	Parameterised circuit trained via gradient descent (mean-squared cost)
		Optimizer	qml.GradientDescentOptimizer(stepsize = 0.2)	Classical optimizer for variational parameters
		Simulation Backend	default.mixed (Pennylane)	Mixed-state simulator used for realistic noise emulation
		Number of Qubits	Depends on features (4–8)	Each qubit encodes one selected feature
7	Cross-validation & runtime control	n_runs	5 independent runs	Average & standard deviation computed for each metric
		StratifiedKFold	n_splits = 3, random_state = 42	Ensures balanced splits for small quantum datasets
8	Evaluation metrics	Accuracy, Precision, Recall, F1 Score	Mean ± Std computed across runs	
		ROC Curve & AUC	roc_curve(), auc()	Evaluates probabilistic separability
		Confusion Matrix	confusion_matrix()	Summarizes classification distribution
9	Explainability (Only for classical models)	SHAP	shap.TreeExplainer(best_model)	Global feature contribution visualization
		LIME	LIME.TabularExplainer(mode = 'classification')	Local interpretability for individual predictions
		Instance Explanation	explain_instance(i = 42)	Explains the 42-nd test sample decision boundary
10	Computational environment	CPU/GPU	Intel i7	Experiments run on CPU; GPU used for Pennylane acceleration
		Software Stack	Python 3.12, Scikit-learn 1.5, Pennylane 0.38, Imbalanced-learn 0.12	Reproducible library versions
11	Performance output	Metrics aggregation	Mean ± Std over 5 runs	Reported in tables and ROC/CM figures
		Result files	Quantum_Results_Full.xlsx,ml_models_tuned_results.csv	Contain complete numerical benchmarks for all models/datasets

Table 3. List of parameters used in the Experiment.

Model	dataset	Sampler	Noise	Accuracy (meanÂ ± std)	Precision (meanÂ ± std)	Recall (meanÂ ± std)	F1 (meanÂ ± std)	Avg Time (s)
SVM	IRIS	SMOTE	0	0.9778 Â ± 0.0157	0.9798 Â ± 0.0143	0.9778 Â ± 0.0157	0.9777 Â ± 0.0158	3.15
SVM	IRIS	SMOTE	0.01	0.9778 Â ± 0.0157	0.9798 Â ± 0.0143	0.9778 Â ± 0.0157	0.9777 Â ± 0.0158	0.74
SVM	IRIS	ADASYN	0	0.9778 Â ± 0.0157	0.9798 Â ± 0.0143	0.9778 Â ± 0.0157	0.9777 Â ± 0.0158	0.6
SVM	IRIS	ADASYN	0.01	0.9889 Â ± 0.0157	0.9899 Â ± 0.0143	0.9889 Â ± 0.0157	0.9889 Â ± 0.0158	0.4
SVM	WINE	SMOTE	0	0.9722 Â ± 0.0227	0.9759 Â ± 0.0189	0.9722 Â ± 0.0227	0.9723 Â ± 0.0225	0.54
SVM	WINE	SMOTE	0.01	0.9722 Â ± 0.0227	0.9759 Â ± 0.0189	0.9722 Â ± 0.0227	0.9723 Â ± 0.0225	0.52
SVM	WINE	ADASYN	0	0.9722 Â ± 0.0227	0.9759 Â ± 0.0189	0.9722 Â ± 0.0227	0.9723 Â ± 0.0225	0.83
SVM	WINE	ADASYN	0.01	0.9722 Â ± 0.0227	0.9759 Â ± 0.0189	0.9722 Â ± 0.0227	0.9723 Â ± 0.0225	0.53
SVM	BREAST_CANCER	SMOTE	0	0.9678 Â ± 0.0109	0.9684 Â ± 0.0112	0.9678 Â ± 0.0109	0.9679 Â ± 0.0110	1.38
SVM	BREAST_CANCER	SMOTE	0.01	0.9649 Â ± 0.0000	0.9663 Â ± 0.0012	0.9649 Â ± 0.0000	0.9650 Â ± 0.0002	1.77
SVM	BREAST_CANCER	ADASYN	0	0.9620 Â ± 0.0041	0.9630 Â ± 0.0035	0.9620 Â ± 0.0041	0.9621 Â ± 0.0040	1.69
SVM	BREAST_CANCER	ADASYN	0.01	0.9591 Â ± 0.0041	0.9600 Â ± 0.0042	0.9591 Â ± 0.0041	0.9592 Â ± 0.0042	2.07
SVM	PIMA	SMOTE	0	0.7251 Â ± 0.0337	0.7278 Â ± 0.0295	0.7251 Â ± 0.0337	0.7256 Â ± 0.0315	6.19
SVM	PIMA	SMOTE	0.01	0.7251 Â ± 0.0245	0.7280 Â ± 0.0208	0.7251 Â ± 0.0245	0.7260 Â ± 0.0228	6.72
SVM	PIMA	ADASYN	0	0.7381 Â ± 0.0272	0.7435 Â ± 0.0240	0.7381 Â ± 0.0272	0.7399 Â ± 0.0257	6.32
SVM	PIMA	ADASYN	0.01	0.7381 Â ± 0.0221	0.7424 Â ± 0.0236	0.7381 Â ± 0.0221	0.7394 Â ± 0.0222	6.92
SVM	HAR	SMOTE	0	0.9567 Â ± 0.0042	0.9568 Â ± 0.0043	0.9567 Â ± 0.0042	0.9566 Â ± 0.0043	63.59
SVM	HAR	SMOTE	0.01	0.9567 Â ± 0.0042	0.9568 Â ± 0.0043	0.9567 Â ± 0.0042	0.9566 Â ± 0.0043	63.52

Table 4. Evaluation results of SVM on all datasets.

Model	Dataset	Sampler	Noise	Accuracy (meanÂ ± std)	Precision (meanÂ ± std)	Recall (meanÂ ± std)	F1 (meanÂ ± std)	Avg Time (s)
KNN	IRIS	SMOTE	0	0.9556 Â ± 0.0157	0.9613 Â ± 0.0119	0.9556 Â ± 0.0157	0.9553 Â ± 0.0160	0.3
KNN	IRIS	SMOTE	0.01	0.9778 Â ± 0.0157	0.9798 Â ± 0.0143	0.9778 Â ± 0.0157	0.9777 Â ± 0.0158	0.39
KNN	IRIS	ADASYN	0	0.9556 Â ± 0.0157	0.9613 Â ± 0.0119	0.9556 Â ± 0.0157	0.9553 Â ± 0.0160	0.32
KNN	IRIS	ADASYN	0.01	0.9667 Â ± 0.0000	0.9697 Â ± 0.0000	0.9667 Â ± 0.0000	0.9666 Â ± 0.0000	0.28
KNN	WINE	SMOTE	0	0.9444 Â ± 0.0000	0.9506 Â ± 0.0022	0.9444 Â ± 0.0000	0.9440 Â ± 0.0006	0.33
KNN	WINE	SMOTE	0.01	0.9537 Â ± 0.0131	0.9591 Â ± 0.0110	0.9537 Â ± 0.0131	0.9535 Â ± 0.0132	0.32
KNN	WINE	ADASYN	0	0.9352 Â ± 0.0346	0.9447 Â ± 0.0265	0.9352 Â ± 0.0346	0.9338 Â ± 0.0361	0.31
KNN	WINE	ADASYN	0.01	0.9352 Â ± 0.0346	0.9447 Â ± 0.0265	0.9352 Â ± 0.0346	0.9338 Â ± 0.0361	0.46
KNN	BREAST_CANCER	SMOTE	0	0.9561 Â ± 0.0072	0.9577 Â ± 0.0075	0.9561 Â ± 0.0072	0.9562 Â ± 0.0073	0.38
KNN	BREAST_CANCER	SMOTE	0.01	0.9532 Â ± 0.0041	0.9549 Â ± 0.0056	0.9532 Â ± 0.0041	0.9533 Â ± 0.0044	0.32
KNN	BREAST_CANCER	ADASYN	0	0.9094 Â ± 0.0041	0.9182 Â ± 0.0095	0.9094 Â ± 0.0041	0.9104 Â ± 0.0044	0.31
KNN	BREAST_CANCER	ADASYN	0.01	0.9152 Â ± 0.0149	0.9220 Â ± 0.0175	0.9152 Â ± 0.0149	0.9161 Â ± 0.0149	0.32
KNN	PIMA	SMOTE	0	0.6753 Â ± 0.0462	0.6904 Â ± 0.0414	0.6753 Â ± 0.0462	0.6803 Â ± 0.0447	0.4
KNN	PIMA	SMOTE	0.01	0.6991 Â ± 0.0372	0.7165 Â ± 0.0334	0.6991 Â ± 0.0372	0.7043 Â ± 0.0361	0.61
KNN	PIMA	ADASYN	0	0.7056 Â ± 0.0457	0.7304 Â ± 0.0433	0.7056 Â ± 0.0457	0.7115 Â ± 0.0447	0.45
KNN	PIMA	ADASYN	0.01	0.6753 Â ± 0.0323	0.6951 Â ± 0.0302	0.6753 Â ± 0.0323	0.6812 Â ± 0.0315	0.4
KNN	HAR	SMOTE	0	0.9233 Â ± 0.0085	0.9257 Â ± 0.0085	0.9233 Â ± 0.0085	0.9234 Â ± 0.0087	1.62
KNN	HAR	SMOTE	0.01	0.9233 Â ± 0.0120	0.9255 Â ± 0.0117	0.9233 Â ± 0.0120	0.9234 Â ± 0.0122	1.45

Table 5. Evaluation results of KNN on all datasets.

Performance overview of quantum ML models

To assess the effectiveness of quantum machine learning algorithms, we evaluated three quantum models- Quantum SVM, Quantum kNN, and VQC- on the same dataset applied to classical models. Each dataset was preprocessed using feature standardisation, Gaussian noise addition, and synthetic oversampling (SMOTE and ADASYN). Tables 9, 10 and 11 summarizes the results, highlighting the impact of quantum feature mapping, kernel distributions, and entanglement depth on classification performance. Each tables from 9, 10 and 11 includes metrics averaged over multiple runs, along with standard deviations, to ensure robustness.

Visual analysis and model explainability

To support transparency and interpretability, we applied SHAP (Shapley Additive explanations) and LIME (Local Interpretable Model-agnostic Explanations) techniques on selected models trained on the provided datasets. SHAP summary plots revealed that features such as mean radius and worst concavity were dominant predictors in the breast cancer classification task. LIME explanations illustrated local decision boundaries for

Model	Dataset	Sampler	Noise	Accuracy (meanÂ ± std)	Precision (meanÂ ± std)	Recall (meanÂ ± std)	F1 (meanÂ ± std)	Avg Time (s)
RandomForest	IRIS	SMOTE	0	0.9667 Â ± 0.0000	0.9697 Â ± 0.0000	0.9667 Â ± 0.0000	0.9666 Â ± 0.0000	8.89
RandomForest	IRIS	SMOTE	0.01	0.9667 Â ± 0.0000	0.9697 Â ± 0.0000	0.9667 Â ± 0.0000	0.9666 Â ± 0.0000	8.84
RandomForest	IRIS	ADASYN	0	0.9444 Â ± 0.0314	0.9473 Â ± 0.0317	0.9444 Â ± 0.0314	0.9443 Â ± 0.0315	8.78
RandomForest	IRIS	ADASYN	0.01	0.9667 Â ± 0.0000	0.9697 Â ± 0.0000	0.9667 Â ± 0.0000	0.9666 Â ± 0.0000	8.96
RandomForest	WINE	SMOTE	0	0.9907 Â ± 0.0131	0.9916 Â ± 0.0119	0.9907 Â ± 0.0131	0.9908 Â ± 0.0130	9.95
RandomForest	WINE	SMOTE	0.01	0.9907 Â ± 0.0131	0.9916 Â ± 0.0119	0.9907 Â ± 0.0131	0.9908 Â ± 0.0130	9.87
RandomForest	WINE	ADASYN	0	0.9907 Â ± 0.0131	0.9916 Â ± 0.0119	0.9907 Â ± 0.0131	0.9908 Â ± 0.0130	9.25
RandomForest	WINE	ADASYN	0.01	0.9907 Â ± 0.0131	0.9916 Â ± 0.0119	0.9907 Â ± 0.0131	0.9908 Â ± 0.0130	9.5
RandomForest	BREAST_CANCER	SMOTE	0	0.9474 Â ± 0.0215	0.9490 Â ± 0.0223	0.9474 Â ± 0.0215	0.9475 Â ± 0.0216	13.94
RandomForest	BREAST_CANCER	SMOTE	0.01	0.9532 Â ± 0.0180	0.9544 Â ± 0.0189	0.9532 Â ± 0.0180	0.9532 Â ± 0.0183	13.59
RandomForest	BREAST_CANCER	ADASYN	0	0.9561 Â ± 0.0072	0.9585 Â ± 0.0072	0.9561 Â ± 0.0072	0.9563 Â ± 0.0071	14.26
RandomForest	BREAST_CANCER	ADASYN	0.01	0.9444 Â ± 0.0109	0.9471 Â ± 0.0129	0.9444 Â ± 0.0109	0.9447 Â ± 0.0111	14.24
RandomForest	PIMA	SMOTE	0	0.7424 Â ± 0.0153	0.7474 Â ± 0.0070	0.7424 Â ± 0.0153	0.7436 Â ± 0.0128	11.91
RandomForest	PIMA	SMOTE	0.01	0.7468 Â ± 0.0231	0.7482 Â ± 0.0159	0.7468 Â ± 0.0231	0.7463 Â ± 0.0200	13.1
RandomForest	PIMA	ADASYN	0	0.7662 Â ± 0.0191	0.7703 Â ± 0.0208	0.7662 Â ± 0.0191	0.7676 Â ± 0.0197	11.94
RandomForest	PIMA	ADASYN	0.01	0.7662 Â ± 0.0106	0.7758 Â ± 0.0062	0.7662 Â ± 0.0106	0.7687 Â ± 0.0094	14.02
RandomForest	HAR	SMOTE	0	0.9575 Â ± 0.0102	0.9582 Â ± 0.0103	0.9575 Â ± 0.0102	0.9574 Â ± 0.0103	105.26
RandomForest	HAR	SMOTE	0.01	0.9458 Â ± 0.0131	0.9468 Â ± 0.0136	0.9458 Â ± 0.0131	0.9457 Â ± 0.0131	108.26

Table 6. Evaluation results of random forest.

Model	Dataset	Sampler	Noise	Accuracy (meanÂ ± std)	Precision (meanÂ ± std)	Recall (meanÂ ± std)	F1 (meanÂ ± std)	Avg Time (s)
DecisionTree	IRIS	SMOTE	0	0.9556 Â ± 0.0157	0.9576 Â ± 0.0171	0.9556 Â ± 0.0157	0.9555 Â ± 0.0157	0.25
DecisionTree	IRIS	SMOTE	0.01	0.9778 Â ± 0.0157	0.9798 Â ± 0.0143	0.9778 Â ± 0.0157	0.9777 Â ± 0.0158	0.25
DecisionTree	IRIS	ADASYN	0	0.9778 Â ± 0.0157	0.9798 Â ± 0.0143	0.9778 Â ± 0.0157	0.9777 Â ± 0.0158	0.25
DecisionTree	IRIS	ADASYN	0.01	0.9333 Â ± 0.0272	0.9388 Â ± 0.0278	0.9333 Â ± 0.0272	0.9330 Â ± 0.0273	0.28
DecisionTree	WINE	SMOTE	0	0.9537 Â ± 0.0472	0.9556 Â ± 0.0458	0.9537 Â ± 0.0472	0.9539 Â ± 0.0469	0.37
DecisionTree	WINE	SMOTE	0.01	0.9167 Â ± 0.0227	0.9198 Â ± 0.0213	0.9167 Â ± 0.0227	0.9165 Â ± 0.0224	0.38
DecisionTree	WINE	ADASYN	0	0.9074 Â ± 0.0693	0.9147 Â ± 0.0626	0.9074 Â ± 0.0693	0.9076 Â ± 0.0693	0.31
DecisionTree	WINE	ADASYN	0.01	0.9537 Â ± 0.0472	0.9556 Â ± 0.0458	0.9537 Â ± 0.0472	0.9539 Â ± 0.0469	0.3
DecisionTree	BREAST_CANCER	SMOTE	0	0.9327 Â ± 0.0109	0.9351 Â ± 0.0120	0.9327 Â ± 0.0109	0.9325 Â ± 0.0112	0.54
DecisionTree	BREAST_CANCER	SMOTE	0.01	0.9474 Â ± 0.0189	0.9481 Â ± 0.0191	0.9474 Â ± 0.0189	0.9474 Â ± 0.0189	0.62
DecisionTree	BREAST_CANCER	ADASYN	0	0.9415 Â ± 0.0109	0.9448 Â ± 0.0104	0.9415 Â ± 0.0109	0.9419 Â ± 0.0107	0.78
DecisionTree	BREAST_CANCER	ADASYN	0.01	0.9415 Â ± 0.0083	0.9446 Â ± 0.0056	0.9415 Â ± 0.0083	0.9417 Â ± 0.0078	0.59
DecisionTree	PIMA	SMOTE	0	0.6991 Â ± 0.0324	0.6997 Â ± 0.0376	0.6991 Â ± 0.0324	0.6988 Â ± 0.0345	0.44
DecisionTree	PIMA	SMOTE	0.01	0.7468 Â ± 0.0191	0.7465 Â ± 0.0150	0.7468 Â ± 0.0191	0.7451 Â ± 0.0160	0.46
DecisionTree	PIMA	ADASYN	0	0.6710 Â ± 0.0311	0.6824 Â ± 0.0308	0.6710 Â ± 0.0311	0.6751 Â ± 0.0308	0.46
DecisionTree	PIMA	ADASYN	0.01	0.7056 Â ± 0.0110	0.7082 Â ± 0.0120	0.7056 Â ± 0.0110	0.7068 Â ± 0.0114	0.66
DecisionTree	HAR	SMOTE	0	0.9050 Â ± 0.0102	0.9062 Â ± 0.0101	0.9050 Â ± 0.0102	0.9049 Â ± 0.0102	22.11
DecisionTree	HAR	SMOTE	0.01	0.8975 Â ± 0.0054	0.8984 Â ± 0.0060	0.8975 Â ± 0.0054	0.8975 Â ± 0.0054	22.7

Table 7. Evaluation results of Decision Tree on all datasets.

SVM and LR classifiers, validating the importance of top-ranked features and offering insight into individual prediction justifications.

From Figs. 7, 8, 9, 10 and 11, we can see SHAP helped identify which features were most important globally, confirming the model relies on medically significant indicators like mean radius and worst concavity. Meanwhile, LIME provided local explanations for individual predictions, demonstrating how specific feature values influence the classification of benign or malignant. Together, these tools enhance trust in the model by offering global and local interpretability. he SHAP plots typically showed red dots indicating higher feature values., Blue dots indicate lower values and. Horizontal clustering near zero indicated low interaction or low impact while the LIME plot breaks down the prediction into positive (supportive) and negative (opposing) contributions, blue bars push the prediction toward one class (e.g., benign, no disease), and orange bars push it toward the opposite class (e.g., malignant, disease present). We also analyse the quantum kernel distributions. Quantum kernels encode classical data into a high-dimensional Hilbert space, enhancing the separability of complex patterns. The kernel distribution, computed as the squared overlap of quantum states, reveals how well the feature map differentiates between classes: tightly clustered intra-class distributions and well-separated

Model	Dataset	Sampler	Noise	Accuracy (meanÂ ± std)	Precision (meanÂ ± std)	Recall (meanÂ ± std)	F1 (meanÂ ± std)	Avg Time (s)
LogisticRegression	IRIS	SMOTE	0	0.9889 Â ± 0.0157	0.9899 Â ± 0.0143	0.9889 Â ± 0.0157	0.9889 Â ± 0.0158	0.4
LogisticRegression	IRIS	SMOTE	0.01	0.9778 Â ± 0.0157	0.9798 Â ± 0.0143	0.9778 Â ± 0.0157	0.9777 Â ± 0.0158	0.44
LogisticRegression	IRIS	ADASYN	0	0.9889 Â ± 0.0157	0.9899 Â ± 0.0143	0.9889 Â ± 0.0157	0.9889 Â ± 0.0158	0.51
LogisticRegression	IRIS	ADASYN	0.01	0.9778 Â ± 0.0157	0.9798 Â ± 0.0143	0.9778 Â ± 0.0157	0.9777 Â ± 0.0158	0.38
LogisticRegression	WINE	SMOTE	0	0.9815 Â ± 0.0262	0.9846 Â ± 0.0218	0.9815 Â ± 0.0262	0.9816 Â ± 0.0260	0.37
LogisticRegression	WINE	SMOTE	0.01	0.9722 Â ± 0.0227	0.9759 Â ± 0.0189	0.9722 Â ± 0.0227	0.9723 Â ± 0.0225	0.42
LogisticRegression	WINE	ADASYN	0	0.9630 Â ± 0.0131	0.9675 Â ± 0.0098	0.9630 Â ± 0.0131	0.9631 Â ± 0.0129	0.43
LogisticRegression	WINE	ADASYN	0.01	0.9722 Â ± 0.0227	0.9759 Â ± 0.0189	0.9722 Â ± 0.0227	0.9723 Â ± 0.0225	0.51
LogisticRegression	BREAST_CANCER	SMOTE	0	0.9649 Â ± 0.0072	0.9658 Â ± 0.0079	0.9649 Â ± 0.0072	0.9650 Â ± 0.0073	0.54
LogisticRegression	BREAST_CANCER	SMOTE	0.01	0.9678 Â ± 0.0041	0.9688 Â ± 0.0047	0.9678 Â ± 0.0041	0.9679 Â ± 0.0042	0.48
LogisticRegression	BREAST_CANCER	ADASYN	0	0.9503 Â ± 0.0041	0.9526 Â ± 0.0038	0.9503 Â ± 0.0041	0.9505 Â ± 0.0040	0.4
LogisticRegression	BREAST_CANCER	ADASYN	0.01	0.9532 Â ± 0.0041	0.9549 Â ± 0.0056	0.9532 Â ± 0.0041	0.9534 Â ± 0.0043	0.49
LogisticRegression	PIMA	SMOTE	0	0.7749 Â ± 0.0251	0.7868 Â ± 0.0273	0.7749 Â ± 0.0251	0.7780 Â ± 0.0249	0.46
LogisticRegression	PIMA	SMOTE	0.01	0.7771 Â ± 0.0221	0.7895 Â ± 0.0238	0.7771 Â ± 0.0221	0.7803 Â ± 0.0219	0.46
LogisticRegression	PIMA	ADASYN	0	0.7727 Â ± 0.0140	0.7894 Â ± 0.0144	0.7727 Â ± 0.0140	0.7764 Â ± 0.0134	0.37
LogisticRegression	PIMA	ADASYN	0.01	0.7814 Â ± 0.0201	0.7965 Â ± 0.0173	0.7814 Â ± 0.0201	0.7848 Â ± 0.0192	0.4
LogisticRegression	HAR	SMOTE	0	0.9600 Â ± 0.0071	0.9606 Â ± 0.0067	0.9600 Â ± 0.0071	0.9600 Â ± 0.0070	26.67
LogisticRegression	HAR	SMOTE	0.01	0.9542 Â ± 0.0096	0.9547 Â ± 0.0092	0.9542 Â ± 0.0096	0.9543 Â ± 0.0095	26.44

Table 8. Evaluation results of logistic regression.

Model	Dataset	Sampler	Noise	Accuracy (meanÂ ± std)	Precision (meanÂ ± std)	Recall (meanÂ ± std)	F1 (meanÂ ± std)	Avg Time (s)
QKNN	Breast Cancer	ADASYN	OFF	0.870 ± 0.000	0.874 ± 0.000	0.871 ± 0.000	0.870 ± 0.000	0.01
QKNN	Breast Cancer	ADASYN	ON	0.934 ± 0.000	0.936 ± 0.000	0.935 ± 0.000	0.936 ± 0.000	0.01
QKNN	Breast Cancer	SMOTE	OFF	0.949 ± 0.000	0.949 ± 0.000	0.949 ± 0.000	0.949 ± 0.000	0.01
QKNN	Breast Cancer	SMOTE	ON	0.936 ± 0.000	0.938 ± 0.000	0.936 ± 0.000	0.936 ± 0.000	0.01
QKNN	HAR	ADASYN	OFF	0.532 ± 0.000	0.526 ± 0.000	0.526 ± 0.000	0.526 ± 0.000	202.10
QKNN	HAR	ADASYN	ON	0.551 ± 0.000	0.553 ± 0.000	0.316 ± 0.000	0.600 ± 0.000	100.21
QKNN	HAR	SMOTE	OFF	0.343 ± 0.000	0.625 ± 0.000	0468 ± 0.000	0.641 ± 0.000	304.33
QKNN	HAR	SMOTE	ON	0.584 ± 0.000	0.583 ± 0.000	0.516 ± 0.000	0.568 ± 0.000	230.18
QKNN	IRIS	ADASYN	OFF	0.956 ± 0.000	0.961 ± 0.000	0.956 ± 0.000	0.955 ± 0.000	22.36
QKNN	IRIS	ADASYN	ON	0.956 ± 0.000	0.961 ± 0.000	0.956 ± 0.000	0.955 ± 0.000	22.15
QKNN	IRIS	SMOTE	OFF	0.956 ± 0.000	0.961 ± 0.000	0.956 ± 0.000	0.955 ± 0.000	22.12
QKNN	IRIS	SMOTE	ON	0.956 ± 0.000	0.961 ± 0.000	0.956 ± 0.000	0.955 ± 0.000	21.98
QKNN	PIMA	ADASYN	OFF	0.532 ± 0.000	0.526 ± 0.000	0.526 ± 0.000	0.526 ± 0.000	0.01
QKNN	PIMA	ADASYN	ON	0.551 ± 0.000	0.553 ± 0.000	0.816 ± 0.000	0.600 ± 0.000	0.01
QKNN	PIMA	SMOTE	OFF	0.636 ± 0.000	0.625 ± 0.000	0.658 ± 0.000	0.641 ± 0.000	0.01
QKNN	PIMA	SMOTE	ON	0.584 ± 0.000	0.583 ± 0.000	0.816 ± 0.000	0.568 ± 0.000	0.01
QKNN	WINE	ADASYN	OFF	0.924 ± 0.000	0.927 ± 0.000	0.924 ± 0.000	0.924 ± 0.000	107.83
QKNN	WINE	ADASYN	ON	0.879 ± 0.000	0.883 ± 0.000	0.882 ± 0.000	0.878 ± 0.000	109.55
QKNN	WINE	SMOTE	OFF	0.906 ± 0.000	0.909 ± 0.000	0.908 ± 0.000	0.905 ± 0.000	101.3
QKNN	WINE	SMOTE	ON	0.906 ± 0.000	0.909 ± 0.000	0.906 ± 0.000	0.906 ± 0.000	102.71

Table 9. Evaluation results of quantum k NN.

inter-class distributions indicate effective encoding. This analysis offers insights into the discriminative power of various quantum feature maps, guiding the selection of kernels and circuit architectures for optimal quantum-enhanced classification. Quantum kernel distributions for all datasets are illustrated in Fig. 12a–e.

The histogram in Fig. 12 shows the distribution of quantum kernel values for all five benchmark datasets used in the study. Each subplot illustrates the frequency of pairwise kernel similarities obtained after encoding the data using the selected quantum feature map. The distribution highlights how the quantum feature map transforms each dataset into the quantum Hilbert space, revealing differences in quantum concentration spread and separability across datasets.

ROC curve and confusion matrix analysis

Accuracy alone can be misleading, especially in imbalanced datasets (e.g., disease detection). ROC curves provide a threshold-independent performance metric that evaluates the classifier's ability to separate positive

Model	Dataset	Sampler	Noise	Accuracy (meanÂ ± std)	Precision (meanÂ ± std)	Recall (meanÂ ± std)	F1 (meanÂ ± std)	Avg Time (s)
QSVM	Breast Cancer	ADASYN	OFF	0.948 ± 0.000	0.949 ± 0.000	0.948 ± 0.000	0.948 ± 0.000	145.8
QSVM	Breast Cancer	ADASYN	ON	0.947 ± 0.000	0.949 ± 0.000	0.947 ± 0.000	0.947 ± 0.000	147.77
QSVM	Breast Cancer	SMOTE	OFF	0.962 ± 0.000	0.964 ± 0.000	0.962 ± 0.000	0.961 ± 0.000	150.27
QSVM	Breast Cancer	SMOTE	ON	0.987 ± 0.000	0.988 ± 0.000	0.987 ± 0.000	0.987 ± 0.000	147.59
QSVM	HAR	ADASYN	OFF	0.494 ± 0.000	0.493 ± 0.000	0.468 ± 0.000	0.629 ± 0.000	510.03
QSVM	HAR	ADASYN	ON	0.526 ± 0.000	0.515 ± 0.000	0.472 ± 0.000	0.448 ± 0.000	605.37
QSVM	HAR	SMOTE	OFF	0.41 ± 0.017	0.521 ± 0.000	0.474 ± 0.000	0.469 ± 0.000	401.06
QSVM	HAR	SMOTE	ON	0.512 ± 0.000	0.514 ± 0.000	0.447 ± 0.000	0.367 ± 0.000	611.46
QSVM	IRIS	ADASYN	OFF	0.911 ± 0.000	0.916 ± 0.000	0.911 ± 0.000	0.911 ± 0.000	24.28
QSVM	IRIS	ADASYN	ON	0.911 ± 0.000	0.916 ± 0.000	0.911 ± 0.000	0.911 ± 0.000	25.08
QSVM	IRIS	SMOTE	OFF	0.911 ± 0.000	0.916 ± 0.000	0.911 ± 0.000	0.911 ± 0.000	30.81
QSVM	IRIS	SMOTE	ON	0.933 ± 0.000	0.935 ± 0.000	0.933 ± 0.000	0.933 ± 0.000	25.12
QSVM	PIMA	ADASYN	OFF	0.494 ± 0.000	0.493 ± 0.000	0.868 ± 0.000	0.629 ± 0.000	308.06
QSVM	PIMA	ADASYN	ON	0.526 ± 0.000	0.515 ± 0.000	0.872 ± 0.000	0.648 ± 0.000	303.77
QSVM	PIMA	SMOTE	OFF	0.755 ± 0.000	0.721 ± 0.000	0.774 ± 0.000	0.679 ± 0.000	311.06
QSVM	PIMA	SMOTE	ON	0.532 ± 0.000	0.514 ± 0.000	0.947 ± 0.000	0.667 ± 0.000	305.46
QSVM	WINE	ADASYN	OFF	0.924 ± 0.000	0.928 ± 0.000	0.924 ± 0.000	0.924 ± 0.000	119.44
QSVM	WINE	ADASYN	ON	0.848 ± 0.000	0.857 ± 0.000	0.851 ± 0.000	0.850 ± 0.000	119.53
QSVM	WINE	SMOTE	OFF	0.922 ± 0.000	0.925 ± 0.000	0.923 ± 0.000	0.923 ± 0.000	113.94
QSVM	WINE	SMOTE	ON	0.891 ± 0.000	0.898 ± 0.000	0.891 ± 0.000	0.891 ± 0.000	114.44

Table 10. Evaluation results of Quantum SVM.

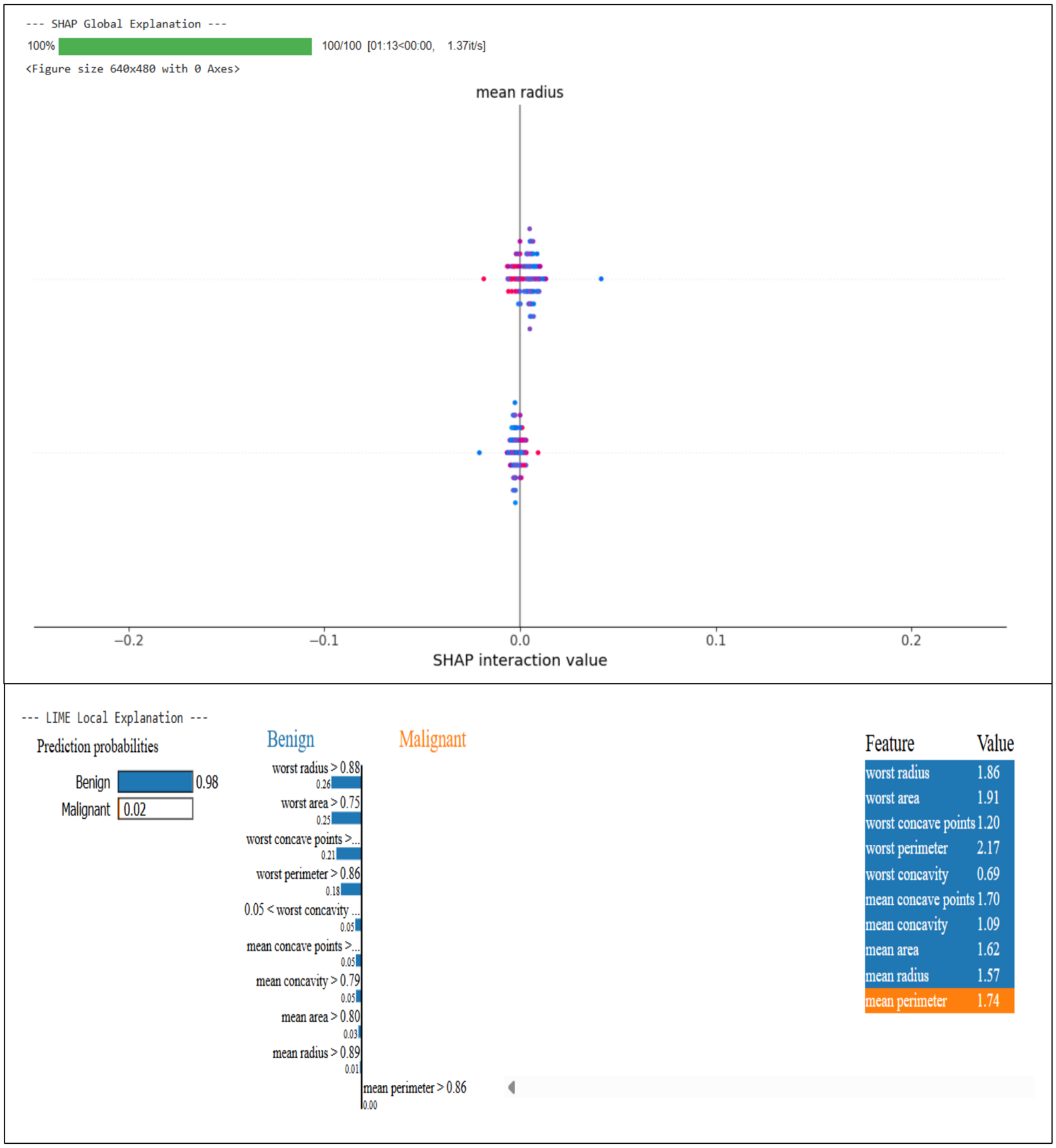
Model	Dataset	Sampler	Noise	Accuracy (meanÂ ± std)	Precision (meanÂ ± std)	Recall (meanÂ ± std)	F1 (meanÂ ± std)	Avg Time (s)
VQC	Breast Cancer	ADASYN	OFF	0.429 ± 0.020	0.416 ± 0.021	0.431 ± 0.012	0.405 ± 0.023	34.41
VQC	Breast Cancer	ADASYN	ON	0.467 ± 0.086	0.482 ± 0.126	0.475 ± 0.087	0.426 ± 0.077	34.39
VQC	Breast Cancer	SMOTE	OFF	0.468 ± 0.038	0.453 ± 0.047	0.468 ± 0.032	0.420 ± 0.033	35.42
VQC	Breast Cancer	SMOTE	ON	0.449 ± 0.038	0.434 ± 0.048	0.449 ± 0.038	0.410 ± 0.033	34.49
VQC	HAR	ADASYN	OFF	0.355 ± 0.000	0.370 ± 0.000	0.316 ± 0.000	0.596 ± 0.000	77.65
VQC	HAR	ADASYN	ON	0.422 ± 0.000	0.418 ± 0.000	0.321 ± 0.000	0.604 ± 0.000	85.42
VQC	HAR	SMOTE	OFF	0.321 ± 0.010	0.351 ± 0.003	0.3763 ± 0.000	0.367 ± 0.003	77.25
VQC	HAR	SMOTE	ON	0.533 ± 0.006	0.456 ± 0.003	0.463 ± 0.000	0.571 ± 0.003	55.94
VQC	IRIS	ADASYN	OFF	0.533 ± 0.073	0.608 ± 0.073	0.533 ± 0.073	0.524 ± 0.073	21.59
VQC	IRIS	ADASYN	ON	0.444 ± 0.109	0.417 ± 0.109	0.444 ± 0.109	0.395 ± 0.109	21.18
VQC	IRIS	SMOTE	OFF	0.444 ± 0.018	0.453 ± 0.018	0.502 ± 0.018	0.418 ± 0.018	21.46
VQC	IRIS	SMOTE	ON	0.393 ± 0.046	0.337 ± 0.046	0.449 ± 0.046	0.358 ± 0.046	21.58
VQC	PIMA	ADASYN	OFF	0.455 ± 0.000	0.470 ± 0.000	0.816 ± 0.000	0.596 ± 0.000	57.65
VQC	PIMA	ADASYN	ON	0.462 ± 0.000	0.478 ± 0.000	0.821 ± 0.000	0.604 ± 0.000	55.01
VQC	PIMA	SMOTE	OFF	0.424 ± 0.006	0.451 ± 0.003	0.763 ± 0.000	0.567 ± 0.003	57.25
VQC	PIMA	SMOTE	ON	0.433 ± 0.006	0.456 ± 0.003	0.763 ± 0.000	0.571 ± 0.003	55.63
VQC	WINE	ADASYN	OFF	0.177 ± 0.043	0.172 ± 0.044	0.177 ± 0.043	0.173 ± 0.044	110.63
VQC	WINE	ADASYN	ON	0.141 ± 0.019	0.141 ± 0.022	0.142 ± 0.019	0.140 ± 0.020	109.94
VQC	WINE	SMOTE	OFF	0.229 ± 0.019	0.229 ± 0.023	0.228 ± 0.020	0.226 ± 0.022	106.22
VQC	WINE	SMOTE	ON	0.198 ± 0.029	0.195 ± 0.033	0.196 ± 0.030	0.196 ± 0.031	106.92

Table 11. Evaluation results of VQC.

and negative classes across all possible thresholds. We analysed the Receiver Operating Characteristic (ROC) curves and Confusion metrix for all models against all datasets, and it is not feasible to present the ROC curve for all cases, so the best ones are presented in Figs. 13, 14, 15, 16 and 17.

Summary of best performing configuration – analysis and interpretation

Table 12 below consolidates the optimal configuration, model type, and data preprocessing technique that yielded the highest classification performance on each of the five datasets, as measured by both Accuracy and execution time.



```
--- SHAP Global Explanation ---
100%|██████████| 100/100 [01:13<00:00,  1.37it/s]
<Figure size 640x480 with 0 Axes>
```

```
--- LIME Local Explanation ---
```

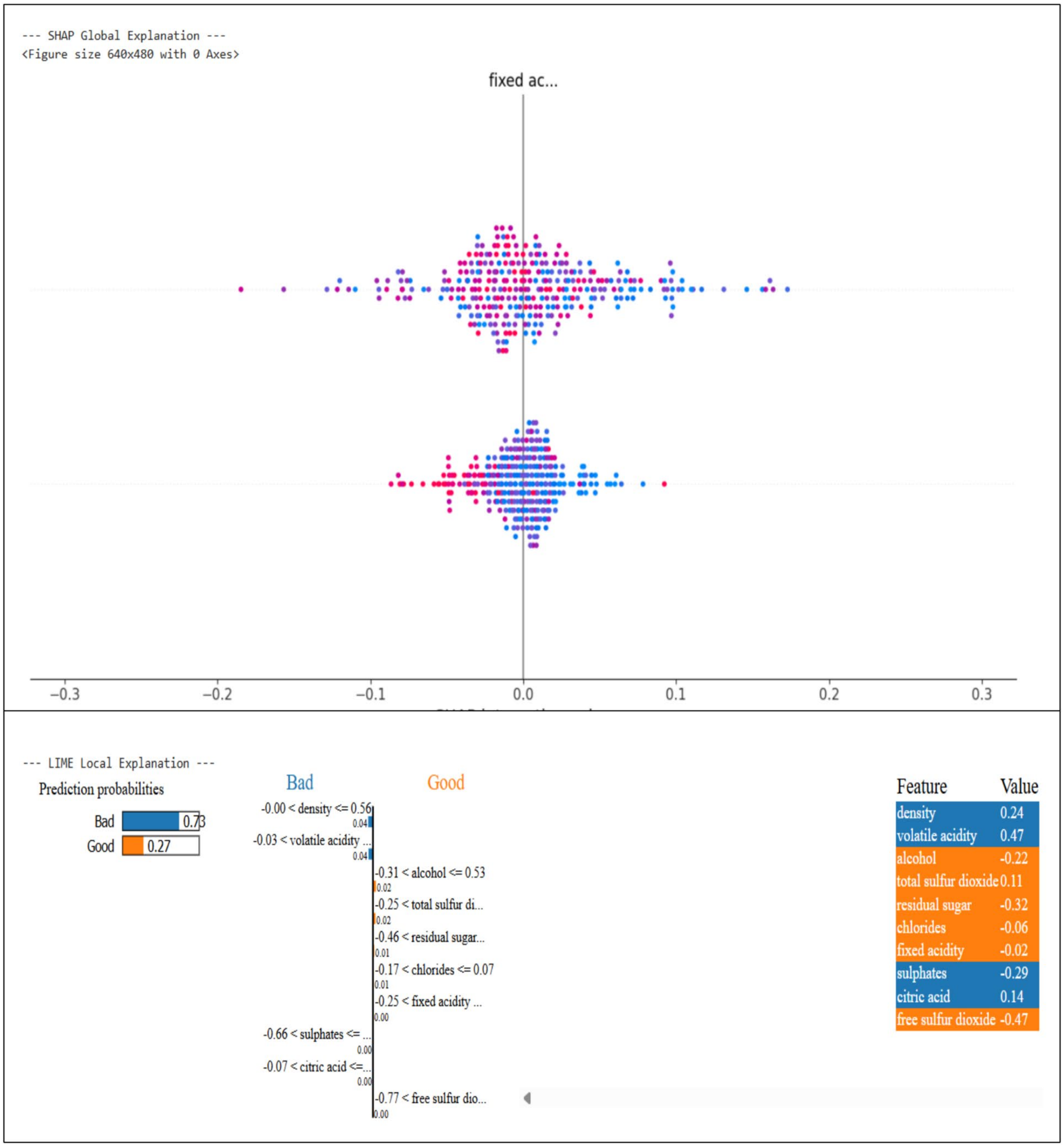
Feature	Value
worst radius	1.86
worst area	1.91
worst concave points	1.20
worst perimeter	2.17
worst concavity	0.69
mean concave points	1.70
mean concavity	1.09
mean area	1.62
mean radius	1.57
mean perimeter	1.74

Fig. 7. SHAP summary plot and LIME local explanation on breast cancer dataset.

The bar chart in Fig. 18 illustrates the performance of the best models for each dataset based on accuracy and execution time. The Logistic Regression achieved a perfect score on the iris, UCI-HAR, and Pima dataset, while QSVM performed strongly on the Breast Cancer, and Random Forest did a tremendous job for wine dataset. The Pima diabetes dataset presented more challenges, with a noticeable drop in the accuracies.

The comparative analysis across multiple benchmark datasets demonstrates a distinct performance pattern between quantum and classical machine learning models. QSVM got a remarkable accuracy (98.7%) when using with the Breast cancer dataset(SMOTE + Noise ON), representing the robustness and capacity of QML to model high-dimensional feature space, but this performance was achieved at significantly higher computational overhead, i.e. 147.59 s, which reflects the simulation capacity of quantum circuits. When evaluating the wine dataset, Random Forest achieved the highest accuracy, 99.7%, illustrating its computational efficiency and superior performance on structured data.

For iris, PIMA and UCI-HAR datasets, Logistic regression consistently performed well with the accuracies of 98.87, 96.00 and 77.5%. With the minimum execution time, it suggests the reliability and computational



```
--- SHAP Global Explanation ---
<Figure size 640x480 with 0 Axes>
```

```
--- LIME Local Explanation ---
```

Feature	Value
density	0.24
volatile acidity	0.47
alcohol	-0.22
total sulfur dioxide	0.11
residual sugar	-0.32
chlorides	-0.06
fixed acidity	-0.02
sulphates	-0.29
citric acid	0.14
free sulfur dioxide	-0.47

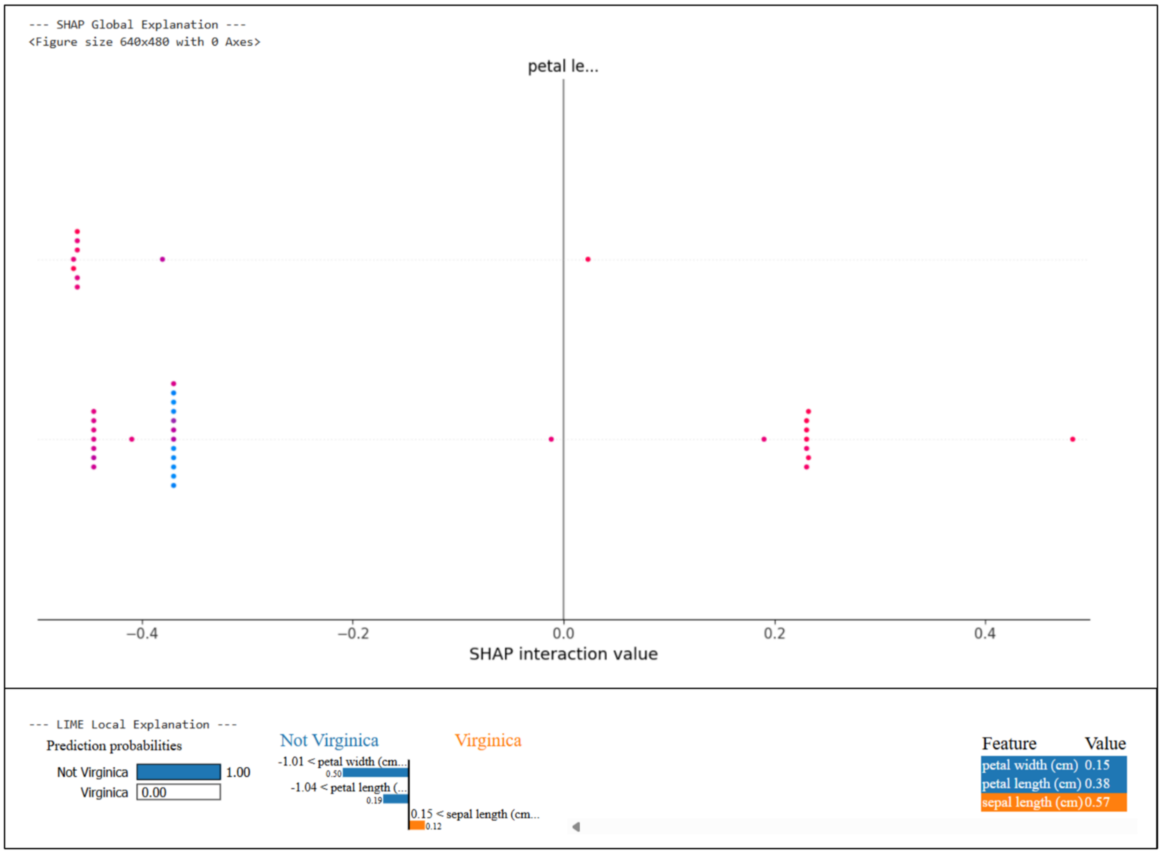
Fig. 8. SHAP summary plot and LIME local explanation on wine quality dataset.

efficiency for complex problems. With the incorporation of SMOTE and ADASYN, it effectively mitigated the class imbalance and enhanced the classification performance.

These findings reveal a clear tradeoff between predictive performance and computational efficiency, highlighting the importance of selecting the right classifier with appropriate methods of preprocessing, i.e., resampling and noise handling, to maximise performance across different domains.To benchmark the results, the study's findings are now compared with existing results in Table 13.

.

The comparison of classification accuracies of existing models reported in the literature with our results is presented in Table 13. For the iris, wine, and Breast cancer datasets, the study's results are significantly better than those of existing studies. However, the results of the study are not so good for the PIMA dataset, but still comparable. There is a lack of work on QkNN with PIMA, UCI HAR, and VQC on UCI HAR datasets, so we fill this gap by including them in our study. The results of Table 13 are shown in Fig. 19 in graphical form.



```
--- SHAP Global Explanation ---
<Figure size 640x480 with 0 Axes>
```

```
--- LIME Local Explanation ---
```

Fig. 9. SHAP summary plot and LIME local explanation on iris dataset.

Insights from the results

The valuable insights of the experiment are presented in Table 14. From the table, it can be reasonably inferred that the classical models delivered fast and consistent results, while the quantum models, particularly QSVM, help capture complex patterns. QkNN can be a reliable option where fast execution time is required. And VQC is very prone to noise with lower stability.

Application relevance and real world impact

The comparative framework developed in this study holds strong relevance for real-world domains characterised by data imbalance, noisy features, and complex non-linear separability. Applications such as fraud detection, medical diagnosis, and industrial IoT anomaly detection often exhibit these challenges, where small but critical minority classes are easily overshadowed by dominant patterns. The demonstrated performance of quantum classifiers, particularly QSVM and VQC, on imbalanced and noisy datasets highlights their potential to enhance decision-making in such high-stakes environments. Quantum feature maps effectively embed overlapping classical data into higher-dimensional Hilbert spaces, allowing improved class separability even with limited training samples. Consequently, the proposed framework establishes a foundation for quantum-enhanced analytics that can be extended to practical sectors once stable and accessible quantum hardware becomes available.

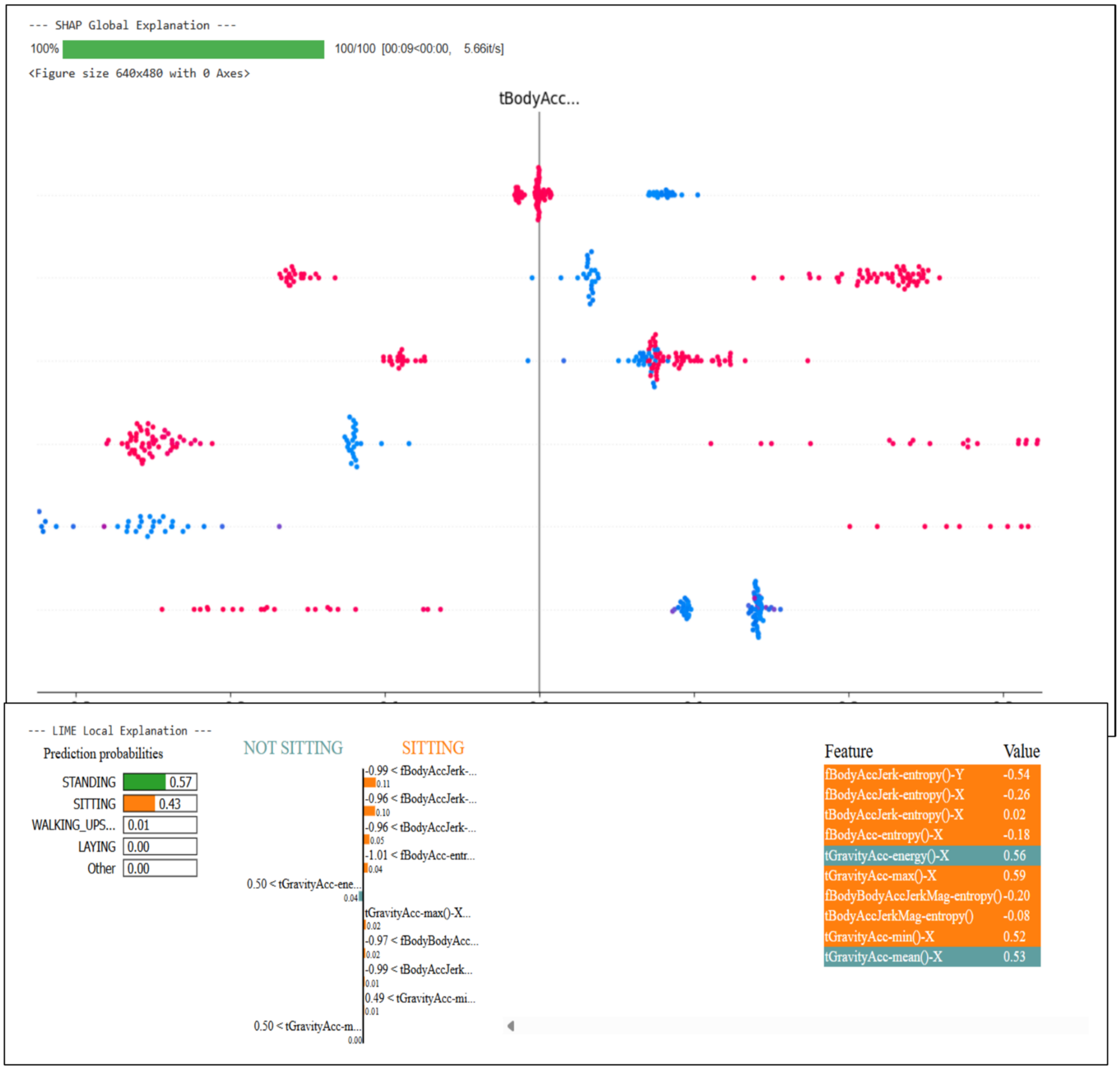
Conclusion and future work

A detailed comparative analysis of classical and quantum machine learning is presented in this study, including five datasets, a range of data preprocessing techniques, i.e. resampling, noise handling and feature selection. The results of the experiment suggested that the performance of ML Models is susceptible to the selection of the preprocessing technique and the type of classifier.

Logistic regression in classical ML Models showcased its performance with the highest accuracy in the Iris, Pima, and UCI-HAR datasets. Similarly, Random Forest achieved the highest accuracy with wine datasets, both with and without noise, demonstrating its potential to remain consistent in all cases.

The quantum-enhanced ML Model QSVM outperforms all other classical and quantum models when implemented using the breast cancer dataset, achieving an accuracy of almost 99%, albeit with the highest execution time, highlighting the limited availability of quantum simulations.

Interpretability tools, such as SHAP and LIME, further validated the trustworthiness of the models by identifying both globally and locally essential features, particularly in medical and activity recognition datasets. Confusion matrix, ROC curve, and analyses complemented traditional metrics, especially under class imbalance,



```
--- SHAP Global Explanation ---
100% 100/100 [00:09<00:00,  5.66it/s]
<Figure size 640x480 with 0 Axes>
```

```
--- LIME Local Explanation ---
```

Prediction probabilities

STANDING	0.57
SITTING	0.43
WALKING_UPS...	0.01
LAYING	0.00
Other	0.00

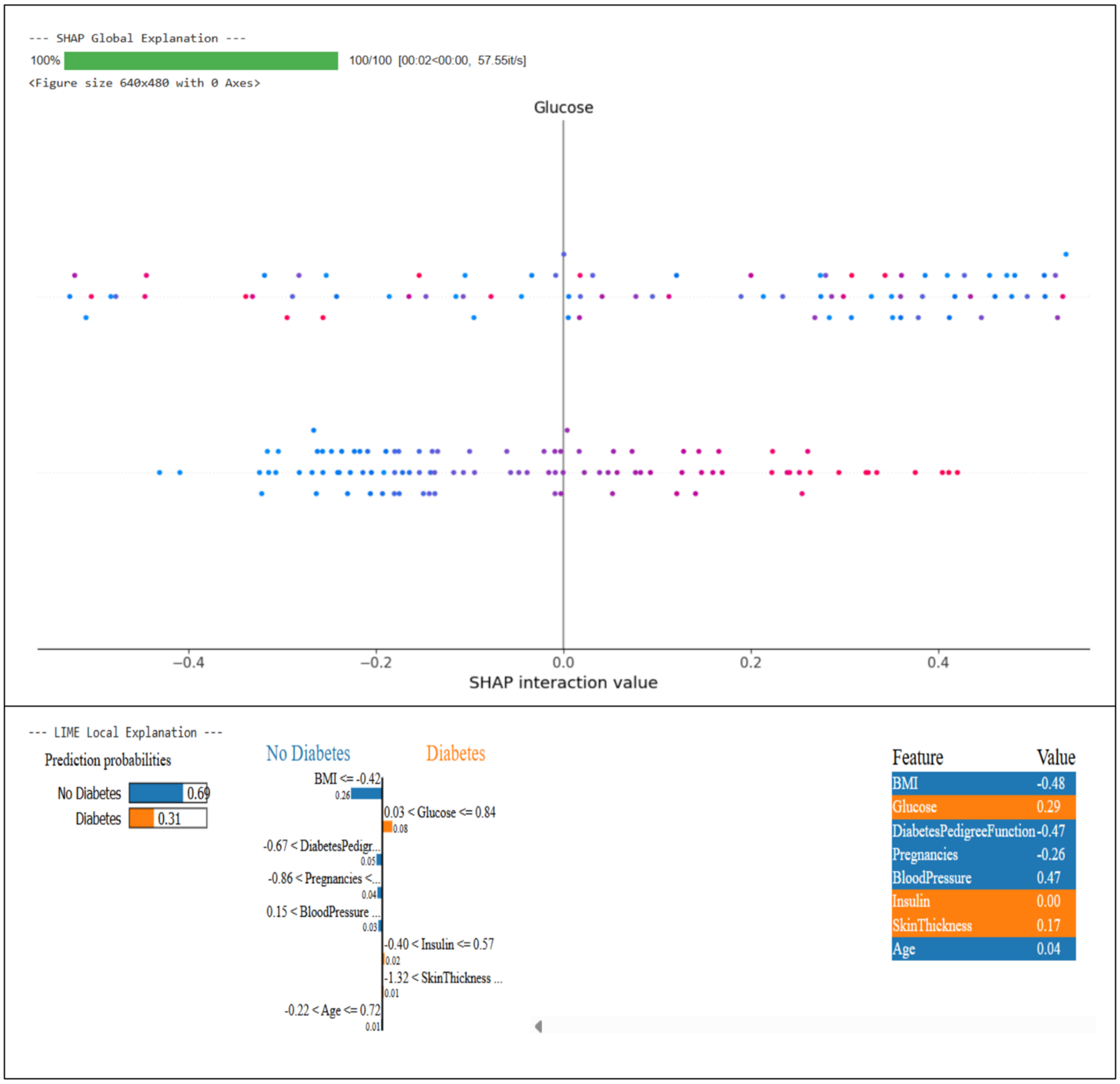
Feature	Value
fBodyAccJerk-entropy()-Y	-0.54
fBodyAccJerk-entropy()-X	-0.26
tBodyAccJerk-entropy()-X	0.02
fBodyAcc-entropy()-X	-0.18
tGravityAcc-energy()-X	0.56
tGravityAcc-max()-X	0.59
fBodyBodyAccJerkMag-entropy()	-0.20
tBodyAccJerkMag-entropy()	-0.08
tGravityAcc-min()-X	0.52
tGravityAcc-mean()-X	0.53

Fig. 10. SHAP summary plot and LIME local explanation on UCI HAR dataset.

highlighting the robust behaviour of the classifier across decision thresholds. Quantum kernel distribution is also analysed to show the similarity values between data points.

When benchmarked against existing literature, the proposed models not only achieved competitive accuracies but also did so using simpler and more interpretable techniques. This suggests that optimal performance does not necessarily require complex architectures; instead, it benefits from judicious model selection, balanced datasets, and effective preprocessing.

In conclusion, this work reinforces the value of hybrid pipelines, which merge classical robustness with emerging quantum techniques, while promoting transparency through explainability tools. These insights can inform the development of future quantum–classical models for real-world classification tasks across healthcare, sensor data, and beyond. Future work will include deploying the optimised quantum circuits on real quantum backends such as IBMQ and ionQ Aria, subject to public availability. This will enable empirical validation of noise resilience and circuit fidelity in a hardware-constrained environment.

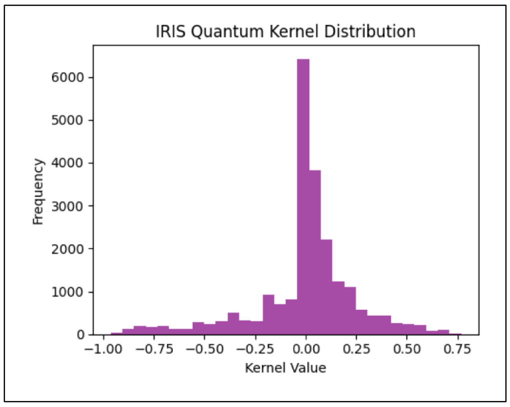


```
--- SHAP Global Explanation ---
100%  100/100 [00:02<00:00, 57.55it/s]
<Figure size 640x480 with 0 Axes>
```

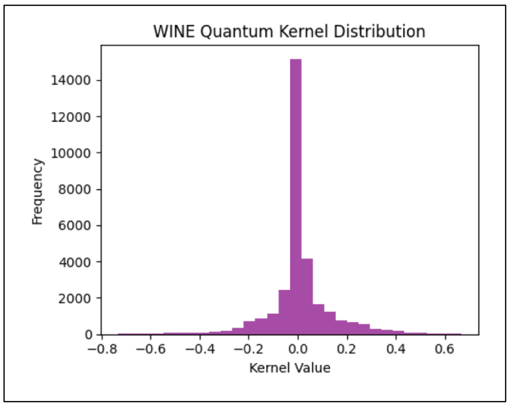
```
--- LIME Local Explanation ---
```

Feature	Value
BMI	-0.48
Glucose	0.29
DiabetesPedigreeFunction	-0.47
Pregnancies	-0.26
BloodPressure	0.47
Insulin	0.00
SkinThickness	0.17
Age	0.04

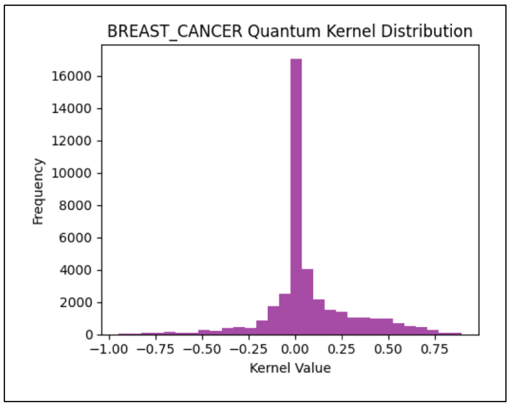
Fig. 11. SHAP summary plot and LIME local explanation on Pima Indian diabetes dataset.



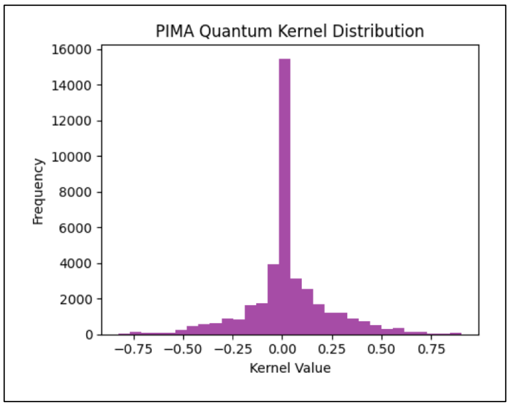
(a) Iris



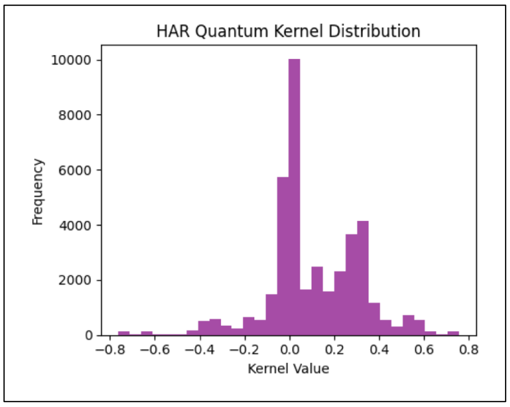
(b) wine



(c) Breast cancer



(d) PIMA



(e) HAR

Fig. 12. Quantum kernel value distributions across datasets.

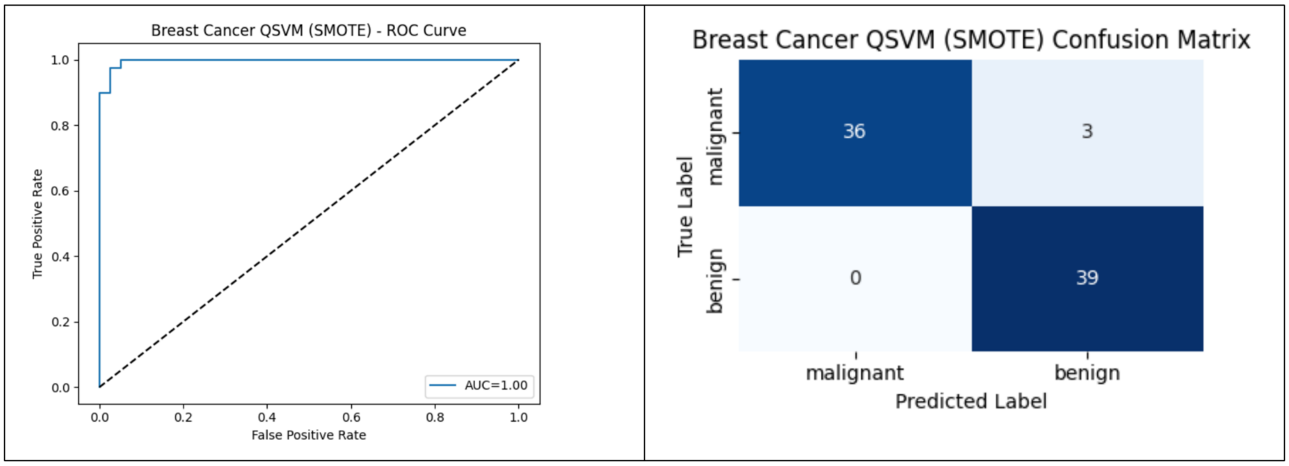


Fig. 13. ROC curve and confusion matrix for QSVM using the breast cancer dataset.

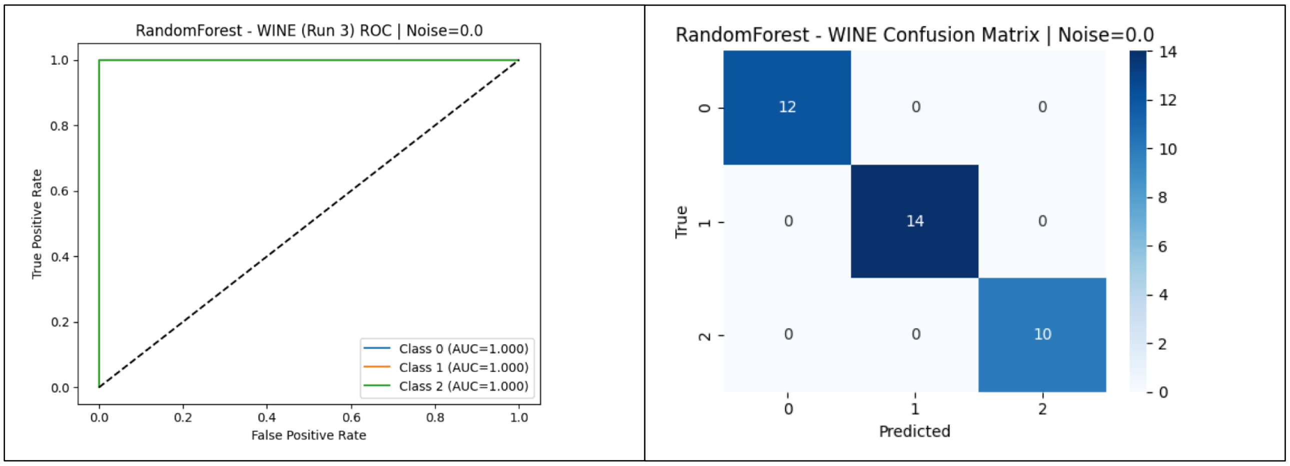


Fig. 14. ROC curve and confusion matrix for RF using the wine dataset.

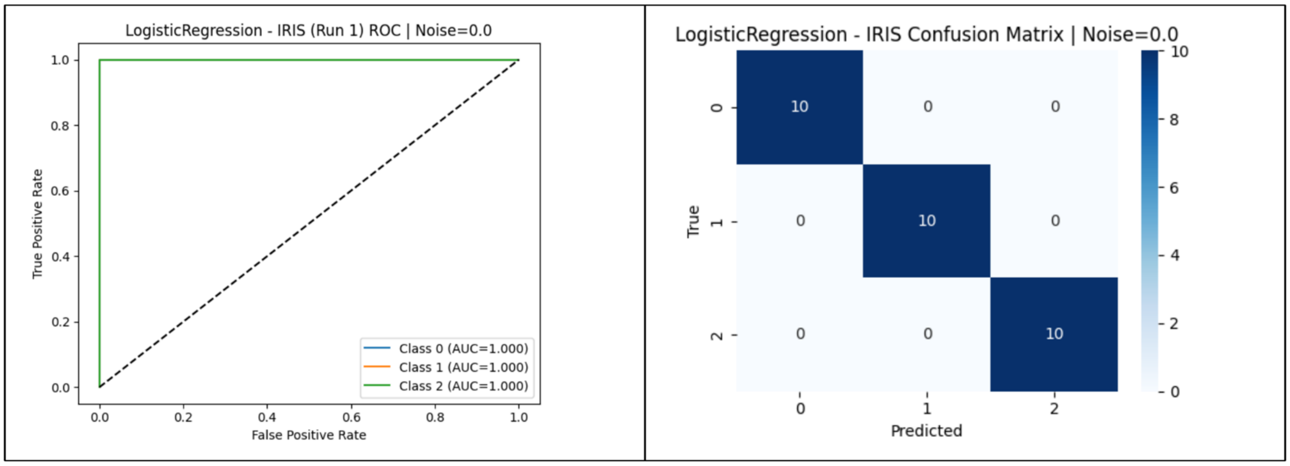
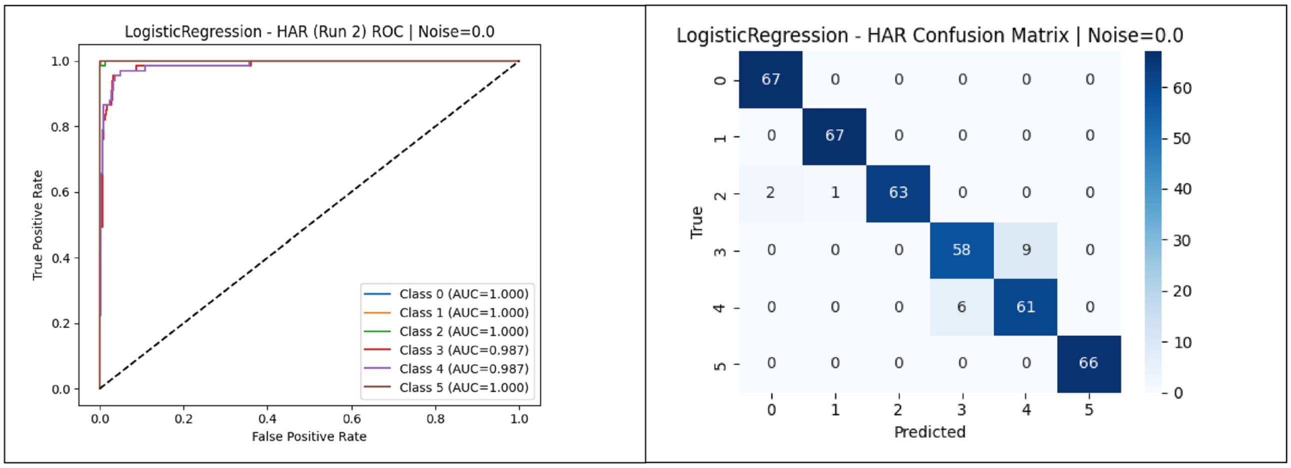
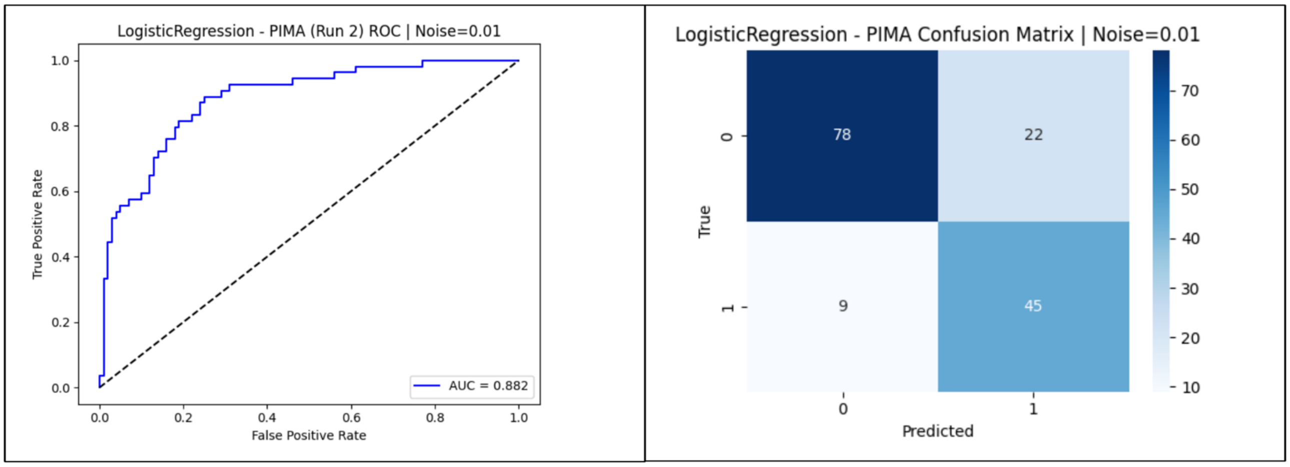


Fig. 15. ROC curve and confusion matrix for LR using the iris dataset.



True \ Predicted	0	1	2	3	4	5
0	67	0	0	0	0	0
1	0	67	0	0	0	0
2	2	1	63	0	0	0
3	0	0	0	58	9	0
4	0	0	0	6	61	0
5	0	0	0	0	0	66

Fig. 16. ROC curve and confusion matrix for LR using the HAR Dataset.



True \ Predicted	0	1
0	78	22
1	9	45

Fig. 17. ROC curve and confusion matrix for LR using the PIMA Dataset.

Dataset	Best- Model	Noise	Sampler	Accuracy	Execution time (s)	Best parameter
Breast Cancer Dataset	Quantum Support Vector Machine	ON	SMOTE	98.7	147.59	Feature map- Angle encoding, c = 1.0
Wine Quality	Random Forest	On/OFF	SMOTE/ADASYN	99.7	9.25	max_depth = None, n_estimators = 50 max_depth = None, n_estimators = 200 max_depth = 10, n_estimators = 100 max_depth = None, n_estimators = 50
Iris	Logistic Regression	OFF	ADASYN	98.87	98.87	C = 10, solver = lbfgs
UCI -HAR Dataset	Logistic Regression	OFF	SMOTE	96.00	26.67	C = 1, solver = liblinear
Pima Indian Diabetes Dataset	Logistic Regression	ON	SMOTE	77.5	0.45	C = 1, solver = liblinear

Table 12. Summary of best performing model.

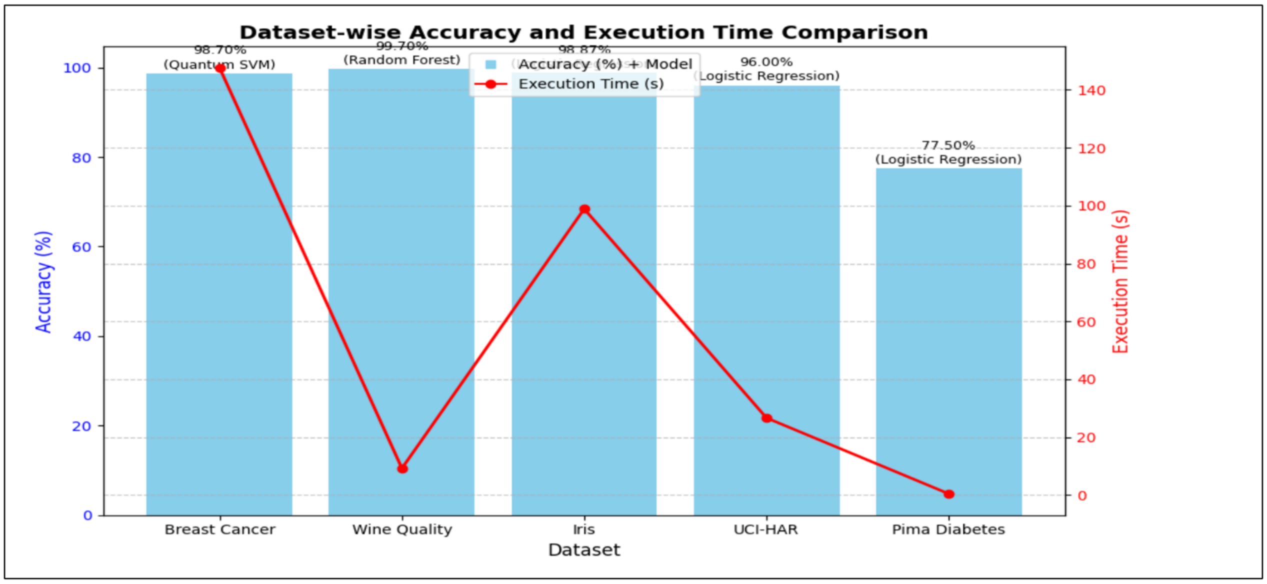


Fig. 18. Graphical representation of the best model performance across datasets.

Reference	Technique	Dataset	Existing results	Our results technique (Accuracy)
[25]	SVM	Iris	98.7 with GA	**98.89**
[26]	Decision Tree	Iris	97	**97.78**
[27]	KNN	Iris	85.71	**97.78**
[28]	LR	Iris	95.55	**98.89**
[29]	RF	Iris	95.55	**95.67**
[30]	QSVM	Iris	99	**93.3**
[31]	QkNN	Iris	95.33	95.6
[32]	VQC	Iris	1.00	53.3
[33]	SVM	Wine	84.56	**97.72**
[34]	Decision Tree	Wine	83.54	**95.37**
[35]	KNN	Wine	59.68	**95.37**
[36]	LR	Wine	76	**96.49**
[36]	RF	Wine	84	**95.37**
[37]	QSVM	Wine	93	**94.4**
[31]	QkNN	Wine	67.98	92.4
[38]	VQC	Wine	53	22.9
[39]	SVM	Breast cancer	78	**96.42**
[40]	DT	Breast cancer	92	**94.74**
[41]	KNN	Breast cancer	94.35	**95.61**
[42]	LR	Breast cancer	84.15	**98.24**
[39]	RF	Breast cancer	95.64	94.7
[37]	QSVM	Breast cancer	90	**98.7**
[43]	QkNN	Breast cancer	98.25	94.9
[44]	VQC	Breast cancer	78	48.8
[45]	SVM	Pima	74.03	73.4
[45]	DT	Pima	72.9	**74.68**
[45]	KNN	Pima	72.9	70.56
[45]	LR	Pima	76.2	**78.14**
[45]	RF	Pima	80.11	74.68
[23]	QSVM	Pima	74	**75.3**
	QkNN	Pima	N/A	63.6
[46]	VQC	Pima	1.00	46.2
[47]	SVM	UCI-HAR	96.9	95.6
N/A	DT	UCI-HAR	N/A	**90.5**
[47]	KNN	UCI-HAR	96.24	92.3
[48]	LR	UCI-HAR	98	96.00
[48]	RF	UCI-HAR	98.47	95.75
[49]	QSVM	UCI-HAR	98	51.2
	QkNN	UCI-HAR	N/A	**58.4**
	VQC	UCI-HAR	N/A	**43.3**

Table 13. Comparison with existing studies.

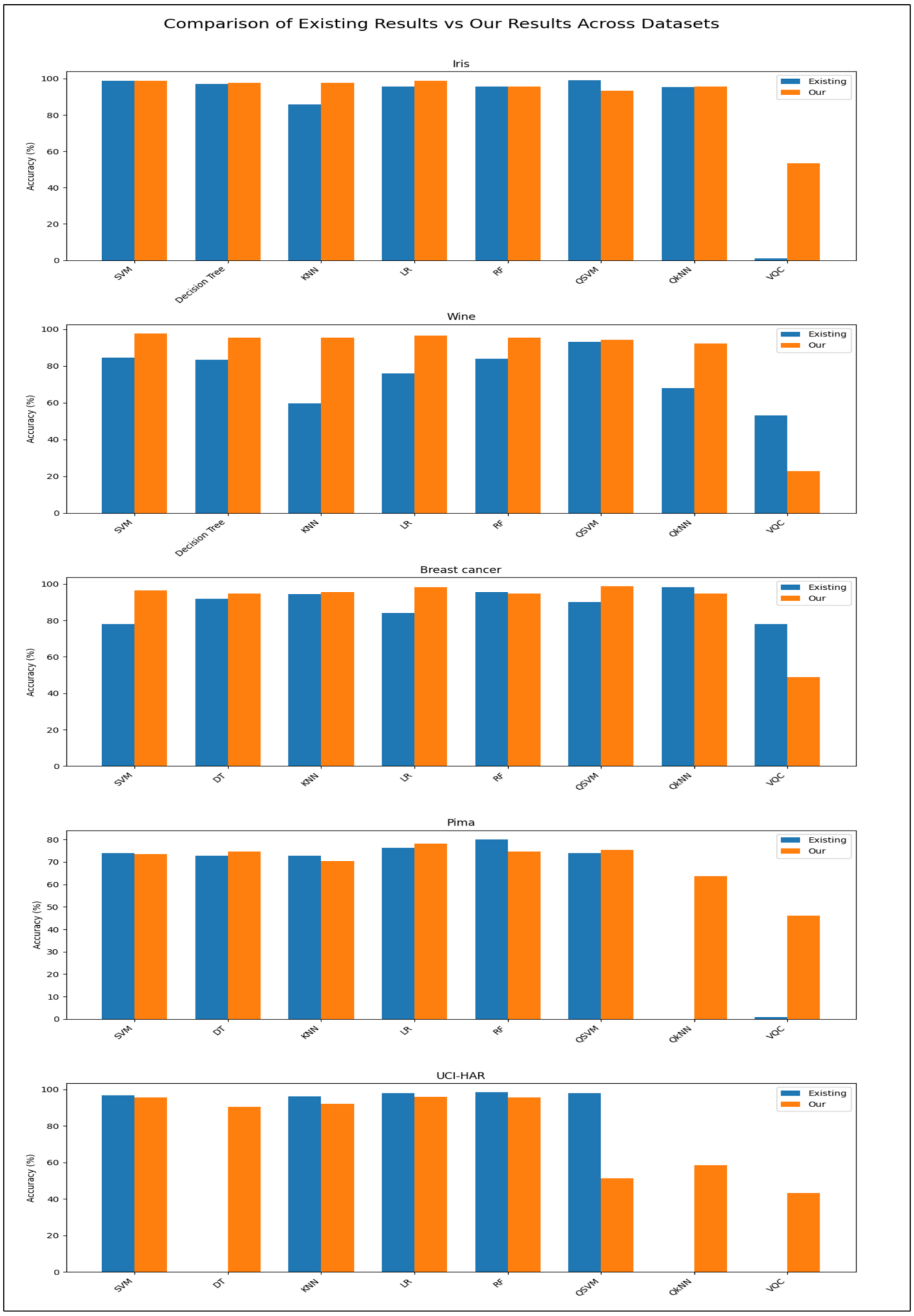


Fig. 19. Comparison of existing vs our techniques results across datasets.

Aspect	Classical ML	QSVM	QKNN	VQC
Accuracy	High, stable	Moderate to high	Moderate	Low, unstable
Sensitivity to noise	Low	Low to moderate	Low	High
Effect of oversampling	Helps ADASYN/SMOTE	Helps SMOTE	Minor effect	Minor effect
Execution time	SVM/RF: medium; KNN: fast	Very high (100–300 s)	Extremely fast (< 1 s)	Moderate (~ 30 s)
Best suited for	Tabular datasets with small/medium features	Datasets where classical SVM may struggle with kernel representations	Fast prototyping	Still experimental; may need deeper circuits or other embeddings

Table 14. Insights from the results.

Data availability

All the datasets are publicly available at UCI and Kaggle - Breast Cancer Wisconsin Dataset - [https://archive.ics.uci.edu/dataset/17/breast+cancer+wisconsin+diagnostic](https:/archive.ics.uci.edu/dataset/17/breast+cancer+wisconsin+diagnostic) , Iris Dataset- [https://archive.ics.uci.edu/dataset/53/iris](https:/archive.ics.uci.edu/dataset/53/iris) , Wine Dataset- [https://archive.ics.uci.edu/dataset/109/wine](https:/archive.ics.uci.edu/dataset/109/wine) , Pima indian diabetes dataset- [https://www.kaggle.com/datasets/uciml/pima-indians-diabetes-database](https:/www.kaggle.com/datasets/uciml/pima-indians-diabetes-database) , UCI human activity recognition dataset- [https://archive.ics.uci.edu/dataset/240/human+activity+recognition+using+smartphones](https:/archive.ics.uci.edu/dataset/240/human+activity+recognition+using+smartphones)

Received: 23 June 2025; Accepted: 10 November 2025
Published online: 05 December 2025

References

1. Mohsin Abdulazeez, A., Zeebaree, D. Q., Abdulqader, D. M. & Zeebaree, D. Q. Machine learning supervised algorithms of gene selection: A review. *Researchgate. Net* **62**(03), 233–244 (2020).
2. Singh, A. & Bist, A. S. A wide scale survey on handwritten character recognition using machine learning. *Int. J. Comput. Sci. Eng.* **7**(6), 124–134. https://doi.org/10.26438/ijcse/v7i6.124134 (2019).
3. C. Havenstein, D. Thomas, S. Chandrasekaran, C. L. Havenstein, and D. T. Thomas, "Comparisons of Performance between Quantum and Classical Machine Learning," *SMU Data Science Review*, vol. 1, no. 4, p. 11, 2018, [Online]. Available: https://scholar.smu.edu/datasciencereviewhttp://digitalrepository.smu.edu.Availableat:https://scholar.smu.edu/datasciencereview/vol1/iss4/11
4. H. K. Gianey and R. Choudhary, "Comprehensive Review On Supervised Machine Learning Algorithms," *Proceedings - 2017 International Conference on Machine Learning and Data Science, MLDS 2017*, vol. 2018-Janua, pp. 38–43, 2017, https://doi.org/10.1109/MLDS.2017.11.
5. C. A. Ul Hassan, M. S. Khan, and M. A. Shah, "Comparison of machine learning algorithms in data classification," *ICAC 2018 - 2018 24th IEEE International Conference on Automation and Computing: Improving Productivity through Automation and Computing*, no. pp. 1–6, 2018, https://doi.org/10.23919/IConAC.2018.8748995.
6. A. Javaid, M. Sadiq, and F. Akram, Skin Cancer Classification Using Image Processing and Machine Learning, *Proceedings of 18th International Bhurban Conference on Applied Sciences and Technologies, IBCAST 2021*, pp. 439–444, 2021, https://doi.org/10.1109/IBCAST51254.2021.9393198.
7. Dornadula, V. N. & Geetha, S. Credit card fraud detection using machine learning algorithms. *Procedia Comput. Sci.* **165**, 631–641. https://doi.org/10.1016/j.procs.2020.01.057 (2019).
8. Karakaya, R. & Kazan, S. handwritten digit recognition using machine learning. *Sakarya Univ. J. Sci.* **25**(1), 65–71. https://doi.org/10.16984/saufenbilder.801684 (2021).
9. P. Vyas, M. R. Fellow, B. P. Rimal, S. Member, and G. Vyas, "Automated Classification of Societal Sentiments on Twitter with Machine Learning," pp. 1–11.
10. O. Arreche, T. Guntur, and M. Abdallah, "XAI-based Feature Selection for Improved Network Intrusion Detection Systems," 2024, *arXiv*: arXiv:2410.10050. https://doi.org/10.48550/arXiv.2410.10050.
11. "Machine Fault Diagnosis Using Audio Sensors Data and Explainable AI Techniques-LIME and SHAP." [Online]. Available: https://www.researchgate.net/profile/Aniqua-Zereen-2/publication/383320912_Machine_Fault_Diagnosis_Using_Audio_Sensors_Data_and_Explainable_AI_Techniques-LIME_and_SHAP/links/66da9385fa5e11512ca21e81/Machine-Fault-Diagnosis-Using-Audio-Sensors-Data-and-Explainable-AI-Techniques-LIME-and-SHAP.pdf
12. Shojaeinasab, A., Jalayer, M., Baniasadi, A. & Najjaran, H. Unveiling the black box: A unified XAI framework for signal-based deep learning models. *Machines* **12**(2), 121. https://doi.org/10.3390/machines12020121 (2024).
13. Liu, C., Han, D., Zhang, X. & Li, N. Research on feature extraction of underwater acoustic target radiation noise based on machine learning algorithm. *J. Phys. Conf. Ser.* **2644**(1), 012008. https://doi.org/10.1088/1742-6596/2644/1/012008 (2023).
14. Franco, G.S., Mahlow, F., Prado, P.M., Pexe, G.E., Rattighieri, L.A. and Fanchini, F.F. Quantum Phases Classification Using Quantum Machine Learning with SHAP-Driven Feature Selection," (2025), *arXiv*: arXiv:2504.10673. https://doi.org/10.48550/arXiv.2504.10673.
15. Kottahachchi-Kankanamge-Don, A. and Khalil, I. 2025 QRLaXAI: quantum representation learning and explainable AI," *Quantum Mach. Intell.*, https://doi.org/10.1007/s42484-025-00253-9.
16. Mücke, S., Heese, R., Müller, S., Wolter, M. & Piatkowski, N. Feature selection on quantum computers. *Quantum Mach. Intell.* https://doi.org/10.1007/s42484-023-00099-z (2023).
17. G. Hellstern, V. Dehn, and M. Zaefferer, Quantum computer based Feature Selection in Machine Learning, 2023. Available: http://arxiv.org/abs/2306.10591
18. Mahesh, B. Machine learning algorithms—A review|enhanced reader. *Int. J. Sci. Res.* **9**(1), 381–386. https://doi.org/10.21275/ART20203995 (2020).
19. Liang, J., Qin, Z., Xiao, S., Ou, L. & Lin, X. Efficient and secure decision tree classification for cloud-assisted online diagnosis services. *IEEE Trans. Dependable Secure Comput.* **18**(4), 1632–1644. https://doi.org/10.1109/TDSC.2019.2922958 (2021).
20. A. Kariya and B. K. Behera, "Investigation of Quantum Support Vector Machine for Classification in NISQ era," pp. 1–15, 2021.
21. Xing, W. & Bei, Y. Medical health big data classification based on KNN classification algorithm. *IEEE Access* **8**, 28808–28819. https://doi.org/10.1109/ACCESS.2019.2955754 (2020).

22. Dang, Y., Jiang, N., Hu, H., Ji, Z. & Zhang, W. Image classification based on quantum K-Nearest-Neighbor algorithm. *Quantum Inf. Process.* https://doi.org/10.1007/s11128-018-2004-9 (2018).
23. Maheshwari, D., Sierra-Sosa, D. & Garcia-Zapirain, B. Variational quantum classifier for binary classification: Real vs synthetic dataset. *IEEE Access* **10**, 3705–3715. https://doi.org/10.1109/ACCESS.2021.3139323 (2022).
24. Maragkopoulos, G., Mandilara, A., Tsili, A. & Syvridis, D. Enhancing the performance of variational quantum classifiers with hybrid autoencoders. *Quantum Inf. Process* **24**(8), 244. https://doi.org/10.1007/s11128-025-04864-w (2025).
25. Hussain, Z. F. et al. A new model for iris data set classification based on linear support vector machine parameter's optimization. *Int. J. Electr. Comput. Eng.* **10**(1), 1079–1084. https://doi.org/10.11591/ijece.v10i1.pp1079-1084 (2020).
26. Aziz, C. F. & Awrahman, B. J. Prediction model based on iris dataset via some machine learning algorithms. *J. Kufa Math. Comput.* **10**(2), 64–69. https://doi.org/10.31642/jokmc/2018/100210 (2023).
27. Pandey, A. & Jain, A. Comparative analysis of KNN algorithm using various normalization techniques. *Int. J. Comput. Netw. Inf. Secur.* **9**(11), 36–42. https://doi.org/10.5815/ijcnis.2017.11.04 (2017).
28. Zhou, Y. Study for iris classification based on multiple machine learning models. *HSET* **23**, 342–349. https://doi.org/10.54097/hset.v23i.3620 (2022).
29. Y. Fakir, Y. Lakhdoura, and R. Elayachi, Comparative Analysis of Random Forest and J48 Classifiers for 'IRIS' Variety Prediction, 2020.
30. J. Yi, K. Suresh, A. Moghiseh, and N. Wehn, Variational Quantum Linear Solver enhanced Quantum Support Vector Machine 2023, *arXiv*: arXiv:2309.07770. https://doi.org/10.48550/arXiv.2309.07770.
31. Guerrero-Estrada, A.-Y., Quezada, L. F. & Sun, G.-H. Benchmarking quantum versions of the kNN algorithm with a metric based on amplitude-encoded features. *Sci. Rep.* **14**(1), 16697. https://doi.org/10.1038/s41598-024-67392-0 (2024).
32. Akrom, M., Herowati, W. & Setiadi, D. R. I. M. A quantum circuit learning-based investigation: A case study in iris benchmark dataset binary classification. *J. Comput. Theor. Appl.* **2**(3), 355–367. https://doi.org/10.62411/jcta.11779 (2025).
33. Ahammed, B. & Abedin, M. M. Predicting wine types with different classification techniques. *Model. Assist. Stat. Appl.* **13**(1), 85–93. https://doi.org/10.3233/MAS-170420 (2018).
34. Dhaliwal, P., Sharma, S. & Chauhan, L. Detailed study of wine dataset and its optimization. *Int. J. Intell. Syst. Appl.* **14**(5), 35–46. https://doi.org/10.5815/ijisa.2022.05.04 (2022).
35. R. Chandra, Comparison of Data Normalization for Wine Classification Using K-NN Algorithm, *IJIIS: International Journal of Informatics and Information Systems*, vol. 5, no. 4, pp. 175–180, 2022, https://doi.org/10.47738/ijiis.v5i4.145.
36. A. Trivedi and R. Sehrawat, Wine Quality Detection through Machine Learning Algorithms," In *2018 International Conference on Recent Innovations in Electrical, Electronics & Communication Engineering (ICRIEECE)*, Bhubaneswar, India: IEEE, 2018, 1756–1760. https://doi.org/10.1109/ICRIEECE44171.2018.9009111.
37. A. Shah, M. Shah, and P. Kanani, Leveraging Quantum Computing for Supervised Classification, In *2020 4th International Conference on Intelligent Computing and Control Systems (ICICCS)*, Madurai, India: IEEE, 2020 256–261. https://doi.org/10.1109/ICICCS48265.2020.9120975.
38. Kölle, M., Giovagnoli, A., Stein, J., Mansky, M.B., Hager, J. and Linnhoff-Popien, C. Improving convergence for quantum variational classifiers using weight re-mapping, (2023) https://doi.org/10.48550/arXiv.2212.14807.
39. Vig, L. Comparative analysis of different classifiers for the Wisconsin breast cancer dataset. *OALib* **01**(06), 1–7. https://doi.org/10.4236/oalib.1100660 (2014).
40. Hossin, M. M. et al. Breast cancer detection: an effective comparison of different machine learning algorithms on the Wisconsin dataset. *Bull. Electr. Eng. Inf.* **12**(4), 2446–2456. https://doi.org/10.11591/eei.v12i4.4448 (2023).
41. Assegie, T. A. An optimized K-nearest neighbor based breast cancer detection. *JRC* https://doi.org/10.18196/jrc.2363 (2021).
42. Murugan, S., Kumar, B.M. and Amudha, S. Classification and Prediction of Breast Cancer using Linear Regression, Decision Tree and Random Forest," In *2017 International Conference on Current Trends in Computer, Electrical, Electronics and Communication (CTCEEC)*, Mysore: IEEE, 2017 763–766. https://doi.org/10.1109/CTCEEC.2017.8455058.
43. Ronggon AA, Rahman MS. Performance Analysis and Noise Impact of a Novel Quantum KNN Algorithm for Machine Learning 2025 arXiv https://doi.org/10.48550/arXiv.2505.06441.
44. Desai, U., Kola, K.S., Nikhitha, S., Nithin, G., Raj, G.P. and Karthik, G. Comparison of Machine Learning and Quantum Machine Learning for Breast Cancer Detection," In *2024 International Conference on Smart Systems for applications in Electrical Sciences (ICSSES)*, Tumakuru, India: IEEE, 2024, 1–6. https://doi.org/10.1109/ICSSES62373.2024.10561257.
45. Zargar, O.S., Bhagat, A., Teli, T.A. and Sheikh, S. Early Prediction Of Diabetes Mellitus On Pima Dataset Using Ml And Dl Techniques, vol. 23, no. 1, (2023).
46. Tudisco, A., Volpe, D. and Turvani, G. Multi-VQC: A Novel QML Approach for Enhancing Healthcare Classification, In 2025 *IEEE International Conference on Quantum Software (QSW)*, (2025), 28–34. https://doi.org/10.1109/QSW67625.2025.00013.
47. Shdefat, A. Y., Mostafa, N., Al-Arnaout, Z., Kotb, Y. & Alabed, S. Optimizing HAR systems: Comparative analysis of enhanced SVM and k-NN classifiers. *Int. J. Comput. Intell. Syst.* **17**(1), 150. https://doi.org/10.1007/s44196-024-00554-0 (2024).
48. Nayak, S., Panigrahi, C., Pati, B., Nanda, S. & Hsieh, M.-Y. Comparative analysis of HAR datasets using classification algorithms. *ComSIS* **19**(1), 47–63. https://doi.org/10.2298/CSIS201221043N (2022).
49. P. Agarwal and M. Alam, "Quantum-Inspired Support Vector Machines for Human Activity Recognition in Industry 4.0," In *Proceedings of Data Analytics and Management*, vol. 90, D. Gupta, Z. Polkowski, A. Khanna, S. Bhattacharyya, and O. Castillo, Eds., In Lecture Notes on Data Engineering and Communications Technologies, vol. 90. , Singapore: Springer Nature Singapore, 2022, pp. 281–290. https://doi.org/10.1007/978-981-16-6289-8_24.

Author contributions
All authors contributed equally to every aspect of the study.

Funding
The authors received no financial support for this work.

Declaration

Competing interests
The authors declare no competing interests.

Additional information
Correspondence and requests for materials should be addressed to V.Y.