Research Article

International Journal of Data Science and Applications (JOINDATA) 8(1), 28-44, 2025

Received: 12-Oct-24 Accepted: 5-Feb-25

homepage: https://dergipark.org.tr/tr/pub/joindata

Feature Selection Using Quantum Feature Maps: Performance Analysis of Classical and Quantum Models on the Breast Cancer Dataset

Sevdanur GENÇ[1*]

[1] Department of Computer Engineering, Faculty of Engineering, Balıkesir University, Balıkesir / Türkiye

sevdanur.genc@balikesir.edu.tr

ABSTRACT

In this study, the performance of classical and quantum machine learning models was compared using the Breast Cancer dataset, which consists of diagnostic data aimed at classifying breast tumor types. Breast cancer, being one of the most common and life-threatening cancers in women, requires accurate diagnostic tools for early detection and effective treatment. The primary objective of this study is to evaluate the accuracy of quantum-assisted models through quantum feature selection methods. Initially, classical machine learning algorithms such as Support Vector Machines (SVM), Decision Trees, Random Forests, and Logistic Regression were applied to the dataset for baseline analysis. Subsequently, a quantum feature map was constructed using the Cirq library, enabling feature transformation based on this map. The classification was performed using the SVM model with the quantum-transformed features. The Logistic Regression and SVM models demonstrated the highest performance among classical machine learning models, achieving an accuracy rate of 96.49%, followed by Random Forest at 94.74% and Decision Tree at 92.11%. In the context of quantum feature transformation, the model utilizing the top five selected features achieved an accuracy rate of 94.74%, in contrast to 98.25% for the model trained with all features. These findings underscore the potential of quantum feature maps in enhancing model performance compared to classical techniques. The results suggest that quantum computing may offer significant advantages when integrated into machine learning frameworks, particularly in domains such as medical diagnostics, where high accuracy is crucial.

Keywords: Quantum Machine Learning, Quantum Feature Maps, Feature Selection, Cirq, Hybrid Quantum-Classical Models

* Corresponding Author's email: sevdanur.genc@balikesir.edu.tr

Kuantum Özellik Haritaları Kullanılarak Özellik Seçimi: Göğüs Kanseri Veri Kümesi Üzerinde Klasik ve Kuantum Modellerin Performans Analizi

ÖZ

Bu çalışmada, meme tümörü türlerini sınıflandırmaya yönelik tanısal veriler içeren Breast Cancer veri kümesi kullanılarak klasik ve kuantum makine öğrenmesi modellerinin performansları karşılaştırılmıştır. Kadınlarda en yaygın ve yaşamı tehdit eden kanser türlerinden biri olan meme kanseri, erken teşhis ve etkili tedavi için doğru tanı araçlarına ihtiyaç duymaktadır. Bu bağlamda, çalışmanın temel amacı, kuantum destekli modellerin doğruluk oranlarını kuantum tabanlı öznitelik seçimi yöntemleriyle değerlendirerek analiz etmektir. İlk aşamada, veri kümesine klasik makine öğrenmesi algoritmaları olan Destek Vektör Makineleri (SVM), Karar Ağaçları, Rastgele Ormanlar ve Lojistik Regresyon uygulanarak temel bir analiz gerçekleştirilmiştir. Ardından, Cirq kütüphanesi kullanılarak bir kuantum öznitelik haritası oluşturulmuş ve bu harita doğrultusunda öznitelik dönüşümü yapılmıştır. Dönüştürülmüş öznitelikler ile sınıflandırma işlemi, SVM modeli kullanılarak gerçekleştirilmiştir. Klasik modeller arasında en yüksek doğruluk oranına %96,49 ile Lojistik Regresyon ve SVM modelleri ulaşmış, bunu %94,74 ile Rastgele Orman ve %92,11 ile Karar Ağacı modelleri takip etmiştir. Kuantum öznitelik dönüşümü bağlamında ise, en iyi beş öznitelikle eğitilen model %94,74 doğruluk oranına ulaşırken, tüm özniteliklerle eğitilen model %98,25 doğruluk göstermiştir. Elde edilen bulgular, kuantum öznitelik haritalarının model performansını klasik tekniklere kıyasla artırma potansiyeline sahip olduğunu ortaya koymaktadır. Sonuçlar, özellikle yüksek doğruluğun kritik öneme sahip olduğu tıbbi tanı gibi alanlarda, kuantum hesaplamanın makine öğrenmesi çerçevelerine entegre edilmesi durumunda önemli avantajlar sunabileceğini göstermektedir.

Anahtar Kelimeler: Kuantum Makine Öğrenimi, Kuantum Özellik Haritaları, Özellik Seçimi, Cirq, Hibrit Kuantum-Klasik Modeller

1 Introduction

Breast cancer is one of the most common types of cancer among women, affecting millions worldwide each year and representing a significant public health concern. Early detection and treatment are critical factors that influence the course of the disease. Identifying breast cancer at an early stage substantially improves patient survival rates and the success of treatment [1]. The accuracy of diagnostic tools used in this process plays a pivotal role in detecting the disease during its early stages. Alongside traditional diagnostic methods, machine learning and data science techniques have increasingly been employed in cancer diagnosis, focusing on developing models that achieve high accuracy rates.

Quantum computers leverage the principles of quantum mechanics to perform computations that differ fundamentally from classical computers. Unlike classical bits, representing information as either 0 or 1, quantum bits (qubits) can exist simultaneously in superpositions of both 0 and 1, enabling quantum computers to process complex calculations more efficiently [2]. Quantum programming involves designing algorithms that exploit quantum phenomena, such as superposition and entanglement, to solve problems. Quantum machine learning (QML) [3] applies these quantum algorithms to machine learning tasks, offering the potential for faster data processing and enhanced model performance in specific applications. Hybrid classical-quantum machine learning [4] combines the strengths of both classical and quantum techniques by using quantum algorithms to enhance or accelerate specific aspects of traditional machine learning workflows, creating a synergistic approach for tackling computationally intensive problems.

Machine learning has gained an essential place in the field of data analysis and modeling. In recent

years, quantum computing and quantum machine learning have been investigated as alternatives to classical methods [5]. Quantum machine learning aims to improve data processing and modeling processes using quantum computing principles. In particular, quantum feature maps [6] offer potential advantages in areas such as data transformation and feature selection.

Classical machine learning methods can provide effective results on large data sets. These methods include algorithms such as support vector machines (SVM), decision trees, random forests, and logistic regression [7]. The performance of these algorithms is well understood in studies on various data sets.

Quantum feature maps [8] make it possible to represent data in high-dimensional spaces using quantum circuits. This approach offers new opportunities beyond classical feature transformation methods. There are limited studies on how quantum feature maps affect feature selection and model performance.

This study compares the performances of classical and quantum machine learning models using the Breast Cancer dataset. In particular, the role of quantum feature maps in data transformation and the impact of this transformation on classical machine learning algorithms are analyzed. The study aims to evaluate the potential advantages of quantum feature maps by examining the accuracy performance of quantum and classical methods. The structure of the paper is as follows: First, the data set and methods used will be detailed. Then, the performances of classical and quantum machine learning models will be compared, and the results will be discussed. Finally, in light of the findings, the potential of quantum machine learning and suggestions for future research will be presented.

2 Literature Review

If the studies conducted in the literature in recent years are examined:

In a study by Prajapati et al. in 2023 [9], quantum computing and machine learning techniques, particularly quantum neural networks, dimensionality reduction algorithms, and support vector machines (SVM), are used for breast cancer prediction. Molecular classification and diagnosis techniques of breast cancer are discussed, as well as the effectiveness of these techniques and comparative analyses of different algorithms.

Wang [10], published in 2024, proposes a new method for feature selection (QSVMF) using quantum support vector machines (QSVM) and a multi-objective genetic algorithm. QSVM aims to improve classification accuracy, reduce the number of features selected, and reduce quantum circuit costs. Experiments on a breast cancer dataset show that QSVMF outperforms classical methods. The QSVMF model achieved an accuracy ranging between 95% and 98%.

Patel et al. in 2024 [11], propose a Hybrid Quantum Classical Algorithm (HQCA) for image classification. The model builds a quantum kernel using *ZZFeatureMap* and applies an image-boosting layer to reduce the size of the dataset. The proposed method provides higher accuracy and efficiency than other quantum models.

Patil et al. in 2024 [12], evaluates the performance of various machine learning models in breast cancer detection by comparing Naive Bayes, Random Forest, Support Vector Machines (SVM), Logistic Regression, and Decision Trees, based on a review of 41 publications. The findings indicate that the Random Forest model achieves the highest accuracy due to its ensemble learning technique, making it a promising tool for healthcare professionals. The study provides a comparative analysis of different machine learning approaches, highlighting their advantages and limitations, and demonstrates that the

proposed method outperforms alternative techniques. The study achieved the following accuracy values: Support Vector Machine (SVM) with 93.85%, Decision Tree with 94.73%, Random Forest with 97.36%, and Logistic Regression with 95.61%.

Sidey-Gibbons in 2019 [13], demonstrates the use of machine learning (ML) techniques for cancer diagnosis using descriptions of nuclei from breast masses. Three predictive models General Linear Models (GLMs), Support Vector Machines (SVMs) with a radial basis function kernel, and single-layer Artificial Neural Networks were developed using a publicly available dataset (N = 683), randomly split into evaluation (n = 456) and validation (n = 227) samples. The models achieved high accuracy (94%-96%), sensitivity (97%-99%), and specificity (85%-94%), with the SVM model reaching a maximum accuracy of 96% and an area under the curve (AUC) of 97%. Performance slightly improved when using a voting ensemble, with accuracy reaching 97%, sensitivity 99%, and specificity 95%.

Sharma et al. in 2018 [14], employs machine learning techniques, specifically comparing Support Vector Machines (SVM) and Artificial Neural Networks (ANN), to classify breast, liver, ovarian, and prostate cancers using both standard organ condition data and gene expression data. The findings reveal that the SVM classifier consistently achieves higher accuracy than the ANN classifier, underscoring the potential benefits of machine learning in enhancing diagnostic precision. The study achieved the following performance values: specificity with 98.2%, sensitivity with 93.22%, accuracy with 96.66% for Support Vector Machine (SVM).

3 Materials And Methods

3.1. Technical Requirements

Cirq [15] is an open-source quantum computing library developed by Google. It can be used to design, simulate, and operate quantum circuits.

Cirq allows users to define and build quantum circuits. Users can design circuits using quantum bits (qubits) and quantum gates (gates) [16]. It allows quantum circuits to be run in various simulators. This is useful for testing the correctness of circuits before running them on real quantum computers. Cirq provides integration with Google's quantum computers, specifically Google Quantum AI's quantum processors. This way, designed quantum circuits can be run on real quantum hardware. Cirq allows you to define parametric quantum gates and build parametric circuits using these gates. This is useful for optimizing and tuning quantum algorithms. It offers various tools and functions for analyzing quantum circuits. It provides information about circuits' depth, complexity, and other characteristics.

The Python programming language also supported the study and prepared with the Cirq library on Colab [17] servers. Python's significant libraries (NumPy [18], Matplotlib [19]) were preferred for data processing and visualization.

3.2. Dataset

In this study, the Breast Cancer dataset is used. The dataset is taken from the Scikit-learn library and contains features and tags for breast cancer diagnosis [20], [21]. The dataset consists of 569 samples, with each sample containing 30 features. Features represent characteristics of cell nuclei, and labels indicate whether the cancer is malignant or benign. A random selection of 10 rows from the dataset, which consists of a total of 31 features including the target variable, is presented in Figure 1.

	mean radius	mean texture	mean perimeter	mean area	mean smoothness	mean compactness	mean concavity	mean concave points	mean symmetry	mean fractal dimension	...	worst texture	worst perimeter	worst area	worst smoothness	worst compactness	worst concavity	worst concave points	worst symmetry	worst fractal dimension	target
204	12.47	18.60	81.09	481.9	0.09965	0.10580	0.08005	0.03821	0.1925	0.06373	...	24.64	96.05	677.9	0.14260	0.2378	0.2671	0.10150	0.3014	0.08750	1
70	18.94	21.31	123.60	1130.0	0.09009	0.10290	0.10800	0.07951	0.1582	0.05461	...	26.58	165.90	1866.0	0.11930	0.2336	0.2687	0.17890	0.2551	0.06589	0
131	15.46	19.48	101.70	748.9	0.10920	0.12230	0.14660	0.08087	0.1931	0.05796	...	26.00	124.90	1156.0	0.15460	0.2394	0.3791	0.15140	0.2837	0.08019	0
431	12.40	17.68	81.47	467.8	0.10540	0.13160	0.07741	0.02799	0.1811	0.07102	...	22.91	89.61	515.8	0.14500	0.2629	0.2403	0.07370	0.2556	0.09359	1
540	11.54	14.44	74.65	402.9	0.09984	0.11200	0.06737	0.02594	0.1818	0.06782	...	19.68	78.78	457.8	0.13450	0.2118	0.1797	0.06918	0.2329	0.08134	1
567	20.60	29.33	140.10	1265.0	0.11780	0.27700	0.35140	0.15200	0.2397	0.07016	...	39.42	184.60	1821.0	0.16500	0.8681	0.9387	0.26500	0.4087	0.12400	0
369	22.01	21.90	147.20	1482.0	0.10630	0.19540	0.24480	0.15010	0.1824	0.06140	...	25.80	195.00	2227.0	0.12940	0.3885	0.4756	0.24320	0.2741	0.08574	0
29	17.57	15.05	115.00	955.1	0.09847	0.11570	0.09875	0.07953	0.1739	0.06149	...	19.52	134.90	1227.0	0.12550	0.2812	0.2489	0.14560	0.2756	0.07919	0
81	13.34	15.86	86.49	520.0	0.10780	0.15350	0.11690	0.06987	0.1942	0.06902	...	23.19	96.66	614.9	0.15360	0.4791	0.4858	0.17080	0.3527	0.10160	1
477	13.90	16.62	88.97	599.4	0.06828	0.05319	0.02224	0.01339	0.1813	0.05536	...	21.80	101.20	718.9	0.09384	0.2006	0.1384	0.06222	0.2679	0.07698	1

10 rows × 31 columns

Figure 1: The random selection of 10 rows from the dataset

3.3. Data Preprocessing

Data pre-processing steps were applied to make the data suitable for analysis:

Scaling: Data were scaled using *StandardScaler* and *MinMaxScaler* [22].

StandardScaler is a technique used to standardize data. The purpose of the scaling formula given in Equation 1 is to normalize the data by making the mean of the data 0 and the standard deviation 1. This allows the data to be scaled on the basis of mean and standard deviation.

$$z = \frac{(x - \mu)}{\sigma} \tag{1}$$

here:

x : Original data value

μ : Average of the data set

σ : Standard deviation of the data set

z : Standardized data value

MinMaxScaler is used to convert data to a specific range, usually between 0 and 1. The purpose of the *MinMaxScaler* formula given in Equation 2 is to normalize each data value between minimum and maximum values.

$$x_{scaled} = \frac{(x - x_{min})}{(x_{max} - x_{min})} \tag{2}$$

here:

x : Original data value

x_{min} : Minimum value in the data set

x_{max} : Maximum value in the data set

x_{scaled} : Scaled data value

First, the data were standardized with *StandardScaler*. Then, the data were normalized between 0 and 1 using *MinMaxScaler*.

Separation into Training and Test Sets: The data were divided into 80% training and 20% test sets. This separation was used to evaluate the model performance.

3.4. Model Performance Metrics

In machine learning, various metrics and methods are employed to objectively evaluate the performance of analyzed models [23]. In this study, the performance of machine learning models was assessed using Precision, Recall, F1 Score, and Accuracy metrics. The mathematical formulations of these metrics are presented in Equations (3), (4), (5), and (6), respectively. In these equations, TP denotes True Positive, FP denotes False Positive, TN denotes True Negative, and FN denotes False Negative.

$$Precision = \frac{TP}{TP+FP} \tag{3}$$

$$Recall = \frac{TP}{TP+FN} \tag{4}$$

$$F1\ Score = 2 \times \frac{Precision \times Recall}{Precision+Recall} \tag{5}$$

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \tag{6}$$

These metrics, used to evaluate the performance of machine learning models, are crucial for identifying the strengths and weaknesses of the model in different scenarios.

3.5. Classical Machine Learning Models

Four different classical machine learning models were used in the study:

Support Vector Machines (SVM): It is a classification algorithm that aims to find an optimal hyperplane that separates the data into two classes and defines this plane with the maximum margin using the most distant points from the data. An SVM model was trained with a linear kernel using the *SVC* [24] class. The Support Vector Machine (SVM) model uses a linear kernel as the kernel function. Among the other default hyperparameters of the model, the regularization parameter C is set to 1.0. If a polynomial kernel were used, the degree of the polynomial would be set to 3. Also, the gamma value for the kernel coefficient is set to scale.

Decision Tree: Builds a model in a tree structure that classifies or regresses data on a set of conditions and decisions; each internal node distinguishes the data based on a property, and each leaf node represents a class or value. A decision tree model was created using the *DecisionTreeClassifier* [25] class. By default, the Decision Tree model uses Gini as the splitting criterion. The best splitter is preferred to select the best split. The maximum depth of the model is not specified, and the tree can grow ultimately. Also, the minimum number of samples required to split a node is set to 2.

Random Forest: It is an ensemble method where multiple decision trees come together, and each tree votes; a majority vote of the trees makes a decision, thus increasing the overall accuracy of the model.

A random forest model was trained using the *RandomForestClassifier* [26] class. In the Random Forest model, 100 decision trees are used by default. While Gini is used as the discrimination criterion, the maximum depth of the trees is not specified so that the trees can grow ultimately. The minimum number of samples required to split a node is set to 2. Also, the maximum number of features used to build each tree is calculated by the square root.

Logistic Regression: It is a model used to estimate the probabilities of data and forms a linear decision boundary in classification problems. The model calculates the class probabilities with a sigmoid function and thus determines to which class each observation belongs. A logistic regression model was created using the *LogisticRegression* [27] class. In the Logistic Regression model, the maximum number of iterations was set to 1000. L-BFGS was used as the optimization algorithm among the default hyperparameters, while the regulation parameter C was set to 1.0. L2 regularization is applied as a penalty. Each model was trained on the training set, and the accuracy performance was calculated based on the test set.

3.6. Quantum Feature Maps

Quantum feature maps were defined and implemented using the Cirq library. It is a technique that enables the transformation of classical data into a high-dimensional space of quantum states through quantum circuits. This transformation aims to improve data representation and processing, especially in quantum machine learning algorithms, by utilizing quantum mechanical properties of data. A quantum feature map is a tool used to encode data features over quantum circuits.

The *QuantumFeatureMap* class creates a quantum circuit using a given number of quantum bits (qubits). These circuits are used to implement quantum feature maps. The class usually constructs the circuit using Hadamard gates (H) [28] and CNOT gates (Controlled-NOT) [29]. While Hadamard gates bring qubits into superposition states, CNOT gates create quantum entanglement. Entanglement is a physics phenomenon in which two separate particles behave identically regardless of distance from any point in the universe [30]. Superposition is when a circuit containing more than one source is considered; the total effect of these sources on the circuit is equal to the sum of the effects of each source alone [31].

The *QuantumFeatureMap* class [32] can contain a transformation matrix used to encode data points in quantum circuits and transform them into quantum states. This transformation allows the data to be processed in quantum circuits. This class also defines the feature map using a quantum circuit containing Hadamard and CNOT gates. This map is created for use in data transformation.

Feature Transformation: It is a technique used in data processing and aims to transform data into a specific format or space. In machine learning, feature transformation is often done to make data more suitable for modeling. Especially in quantum machine learning, this transformation is related to the representation of classical data in quantum circuits. The function transforms the data by applying the quantum feature map using a given number of quantum bits (qubits). This allows classical data to be encoded in quantum circuits. The function usually uses a constant matrix for the transformation process. This matrix determines how the data is represented in quantum circuits. By applying the input data to the quantum circuits, the function enables the data to be transformed into quantum states. This makes the data suitable for quantum calculations.

The *quantum_feature_transformation* function transformed the data using a quantum feature map. The SVM model was retrained with the transformed data.

In the Quantum Feature Map model, quantum feature transformation is realized by using 1 qubit for each feature. The circuit is constructed with Hadamard gates (*cirq.H*) and CNOT gates. In this transformation, a constant transformation matrix (*transformation_matrix = np.ones*) was used to fit the data to the quantum circuit. The hyperparameters of the SVM model were left as default, and a simple quantum feature transformation was applied to the quantum circuits with Hadamard and CNOT gates.

3.7. Performance Evaluation

The performance of classical and quantum models is compared:

Classical Models: The accuracy performance of each classical model was calculated on the test set.

Quantum Models*:* The SVM model was trained and tested with quantum feature map transformed data. The accuracy performance of each feature was calculated to evaluate the effect of features.

3.8. Feature Selection

The data set was split into training and test sets. The training and test sets were split into 80% training and 20% testing using the *train_test_split* function. The following steps were applied for model evaluation:

Selection of Best Features: The best features were selected using average accuracy rates. The accuracies of the selected features were visualized, and the top 5 features with the best performance were identified.

Evaluating the Performance of Features: A Support Vector Machine (SVM) model was trained for each feature after quantum transformation. The model's success was measured by its accuracy on the test set. This process was repeated 5 times for each feature, and average accuracy values were calculated.

Comparison of Model Performance: The performances of SVM models trained with all features and only the best features are compared. Accuracy rates are visualized comparatively.

4 Experimental Results

4.1. Performance of Classical Machine Learning Models

This study tested four different classical machine-learning models using the Breast Cancer dataset. The accuracy of these models on all features [33] and their performance metrics are shown in Table 1.

Table 1: Performance Of Classical Machine Learning Models

Model	Acc	Prec	Rec	F1
SVM	0.9561	0.9714	0.9577	0.9645
Decision Tree	0.9298	0.9437	0.9437	0.9437
Random Forest	0.9649	0.9589	0.9859	0.9722
Logistic Regression	0.9737	0.9722	0.9859	0.979

The results presented in the table illustrate the performance of various classical machine learning models on the breast cancer dataset. The SVM model achieved the highest success rate with an accuracy of 95.61%, coupled with a precision of 97.14% and a recall of 95.77%. This indicates a high level of correct positive predictions and a low false positive rate. While demonstrating a lower performance with an

accuracy of 92.98%, the Decision Tree model maintains stable results with a precision and recall of 94.37%. On the other hand, the Random Forest model ranks second with an accuracy of 96.49%, showcasing the highest recall rate of 98.59%, which indicates a strong ability to detect the positive class. Finally, the Logistic Regression model displays robust performance with an accuracy of 97.37%, offering competitive results in both precision (97.22%) and recall (98.59%). Overall, the SVM and Random Forest models present higher overall accuracy and performance metrics compared to other models, highlighting their effectiveness as viable options for breast cancer classification tasks.

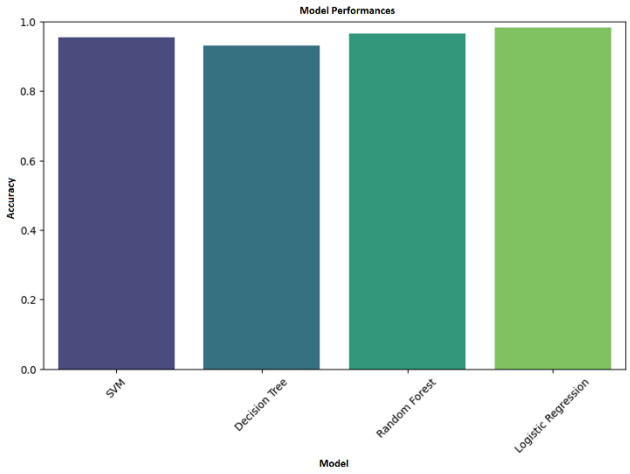

Figure 2: Performance distributions of classical machine learning models

As Figure 3 shows, the accuracy values for each feature are recorded in the *all_feature_accuracies* array. The code is executed a specified number of times (five times) to compute the accuracy of each feature. The *average_accuracies* array obtains the average accuracy value for each feature by calculating the mean of the accuracy values across all conditions. By employing the *np.argsort* function, the indices of the top five features with the highest average accuracy values are determined. The portion [-*num_best_features:*] is utilized to select the five features with the highest accuracy values.

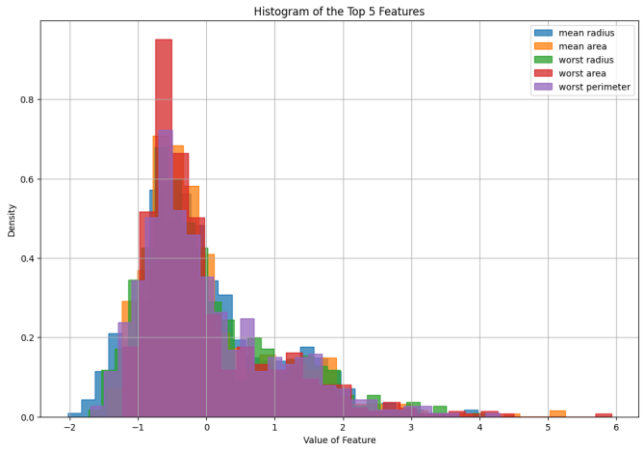

Figure 3: Top five features selected by mean absolute value

Four different classical machine learning models were tested for the five selected features. The accuracy of these models on all features is shown in Table 2.

Table 2: Performance of Classical Machine Learning Models after Top 5 Feature Selection

Model	Acc	Prec	Rec	F1
SVM	0.9649	0.9467	1.0000	0.9726
Decision Tree	0.9211	0.9306	0.9437	0.9371
Random Forest	0.9474	0.9452	0.9718	0.9583
Logistic Regression	0.9649	0.9467	1.0000	0.9726

The analysis evaluated the classification performance of four different machine learning models (SVM, Decision Trees, Random Forests, and Logistic Regression) using the top five selected features. According to the results, both the SVM and Logistic Regression models demonstrated the highest performance with an accuracy of 96.49%. These models were particularly notable for correctly identifying all positive classes (Recall = 1.0) and achieving a high F1 score (0.9726). The Random Forest model also provided satisfactory results with an accuracy of 94.74% and a balanced Precision-Recall ratio. However, the Decision Tree model, with an accuracy of 92.11%, exhibited comparatively lower performance but still delivered an acceptable level of classification success. Overall, the results indicate that the SVM and Logistic Regression models outperform the others on this dataset.

Table 3: Performance Metrics of all features and best features

Model	Acc (Best Features)	Acc (All Features)	Prec (Best Features)	Prec (All Features)	Rec (Best Features)	Rec (All Features)	F1 (Best Features)	F1 (All Features)
SVM	0.9649	0.9561	0.9467	0.9714	1.0000	0.9577	0.9726	0.9645
Decision Tree	0.9211	0.9298	0.9306	0.9437	0.9437	0.9437	0.9371	0.9437
Random Forest	0.9474	0.9649	0.9452	0.9589	0.9718	0.9859	0.9583	0.9722
Logistic Regression	0.9649	0.9737	0.9467	0.9722	1.0000	0.9859	0.9726	0.979

The performance metrics of classical machine learning models using both the best-selected features and all available features are summarized in Table 3. The results indicate that Logistic Regression achieved the highest accuracy (97.37%) when trained on all features, followed closely by Random Forest (96.49%) and SVM (95.61%). When using only the best-selected features, SVM and Logistic Regression both achieved an accuracy of 96.49%, demonstrating the effectiveness of feature selection. Precision and recall values show that SVM and Logistic Regression exhibited the highest recall (1.000) when trained on the best features, whereas the highest precision was observed in Logistic Regression (97.22%) and SVM (97.14%) when using all features. The F1 scores further confirm that the overall best-performing models were Logistic Regression (0.979) and Random Forest (0.9722) when trained on all features, with a slight decrease when using selected features. These findings suggest that while feature selection maintains competitive performance, training models with all features leads to superior classification accuracy, highlighting the potential impact of feature selection on model generalization.

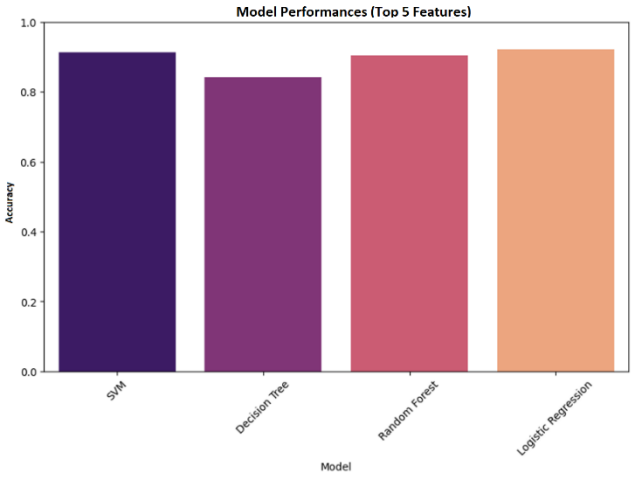

Figure 4: Performance distributions of classical machine learning models

As Table 4 shows, the analysis of feature significance revealed that the selected features play a crucial role in model performance. Among the top five features, the "worst perimeter" exhibited the highest accuracy at 95.61%, indicating its strong predictive power in distinguishing between classes. Following closely, both "worst radius" and "worst area" achieved an accuracy of 94.74%, suggesting their relevance in the classification process. The "mean area" and "mean radius" features also contributed positively, with accuracies of 93.86% and 92.98%, respectively. Overall, these findings underscore the importance of these specific features in enhancing the accuracy of the classification models, thereby demonstrating their potential utility in breast cancer diagnosis.

Table 4: Accuracy Rates of Top 5 Feature Selection

Feature Index	Accuracy	Average Accuracy
0	mean radius	0.9298
3	mean area	0.9386
20	worst radius	0.9474
23	worst area	0.9474
22	worst perimeter	0.9561

Considering Figure 4 and Table 2, the Logistic Regression and SVM model achieved the highest accuracy rate (96.49%), followed by the Random Forest and Decision Tree models.

4.2. Application of Quantum Feature Maps

After applying the quantum feature transformation, the effect of each feature on the model performance was evaluated. Figure 5 shows the average accuracy of each feature. Table 5 and Figure 6 show the values of the best five features.

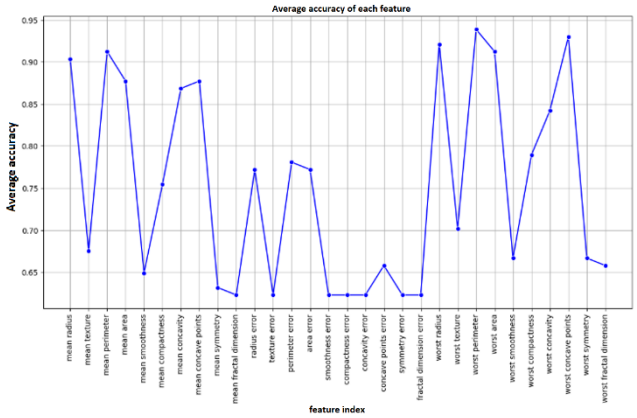

Figure 5: Average accuracy of each feature

The worst perimeter feature has the highest accuracy rate, while the mean perimeter and worst area features have equal accuracy rates.

Table 5: Accuracy Rates of Top 5 Feature Selection

Feature Index	Accuracy	Average Accuracy
2	mean perimeter	0.9123
23	worst area	0.9123
20	worst radius	0.9211
27	worst concave points	0.9298
22	worst perimeter	0.9386

Table 6 shows the performance metric results of the model on all features and the best five features.

Table 6: Performance Metrics of all features and best features

Performance Metrics	All Features	Best Features
Accuracy	0.9825	0.9474
Precision	0.9726	0.9452
Recall	1.0	0.9718
F1 Score	0.9861	0.9583

As Table 6 shows, the performance metrics presented in the table highlight the impact of feature selection on model efficacy. When utilizing all features, the model achieved an accuracy of 0.9825, which indicates a high level of overall correctness in predictions. However, when evaluated with the selected best features, the accuracy decreased to 0.9474. This suggests that while the full feature set may provide a more nuanced understanding of the data, the selected features still maintain a substantial predictive capability. Precision scores exhibited a similar trend, with values of 0.9726 for all features and 0.9452 for the best features, reflecting a decrease in the proportion of true-positive identifications. Notably, the recall metric remained remarkably high, at 1.0 for all features, and only slightly reduced to 0.9718 with the best features, indicating that the model retained its effectiveness in identifying actual positive cases despite the reduction in features. The F1 Score, a harmonic mean of precision and recall, also demonstrated this trend, decreasing from 0.9861 to 0.9583. Overall, these metrics underscore the importance of feature selection, revealing that while fewer features can lead to a decline in some performance aspects, the model still retains strong predictive abilities with the best features selected.

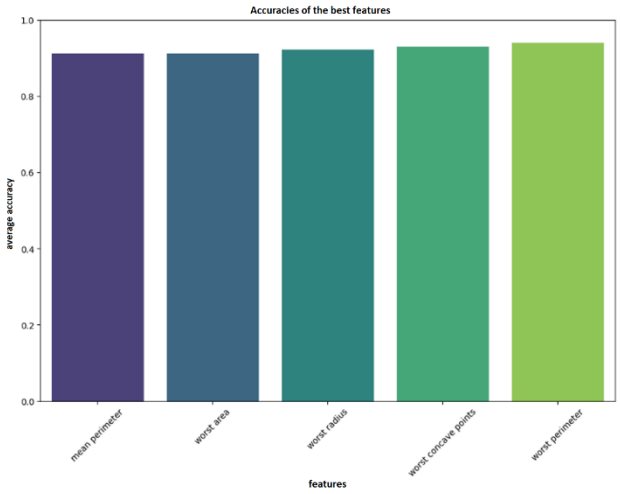

Figure 6: Accuracy distributions of the top 5 features

Accuracy rates of the features play an important role in increasing the success of the model.

4.3. Comparative analysis of the models

In this study, the performances of classical machine learning models on breast cancer datasets were analyzed, and the best features were selected by applying quantum feature transformation. The results obtained are as follows:

Performance of Classical Machine Learning Models (Top 5 best feature selection and mean absolute value): Logistic Regression and SVM models were the most successful models, with a 96.49% accuracy rate. It was followed by Random Forest with a 94.74% accuracy rate and Decision Tree with a 92.11% accuracy rate.

Quantum Feature Transformation and Selection of the Best Features: The best five features were determined from quantum feature transformation. The accuracy rate of the model trained with these features was 94.74%. However, the accuracy rate of the model trained with all features was calculated as 98.25%.

5 Conclusion, Discussion and Suggestions

It has been observed that quantum feature maps offer potential advantages, especially in high-dimensional data transformations. Quantum transformation has been found to provide results beyond classical methods in some cases and be influential in selecting features. However, the current limitations and computational costs of quantum computing must also be considered.

5.1. Conclusion

This study compared the performances of classical and quantum machine learning approaches using the Breast Cancer dataset. Classical machine learning models (SVM, Decision Trees, Random Forests, Logistic Regression) provided high accuracy rates on the dataset and generally performed as expected. Each of these models offered effective results in classifying the data.

The study's findings compare the performances of classical and quantum feature selection methods on breast cancer datasets. When the best five features selected by the classical method were used, the

Logistic Regression and SVM model was the most successful model, with an accuracy of 96.49%. However, the model trained with the best five features selected using quantum feature transformation reached 94.74% accuracy. The model trained with all features performed most with 98.25% accuracy.

These results show that quantum feature transform can positively affect feature selection and model performance. In particular, the quantum method provides an advantage over classical methods by enabling high accuracy rates to be achieved using fewer features. Quantum machine learning methods have presented important findings, especially regarding the use of quantum feature maps. The transformation process using quantum feature maps improved the accuracy performance of some features. In particular, the selection of the best features after quantum transformation increased the model's accuracy and, in some cases, provided higher performance than classical methods. Classification with the best features improved the accuracy rate, which shows that quantum methods can be advantageous in some data sets.

5.2. Discussion

In this study, the performance of classical and quantum feature selection methods on breast cancer datasets was compared with findings from the literature. Wang (2024) reported that the QSVMF model achieved accuracy between 95% and 98%, while Patil et al. (2024) obtained accuracy values of 93.85% for SVM, 94.73% for Decision Tree, 97.36% for Random Forest, and 95.61% for Logistic Regression. Similarly, Sidey-Gibbons (2019) demonstrated high classification performance, with SVM achieving a maximum accuracy of 96% and an AUC of 97%, further improving to 97% accuracy in a voting ensemble. Sharma et al. (2018) also reported high SVM performance, with specificity of 98.2%, sensitivity of 93.22%, and accuracy of 96.66%. In comparison, our study found that when the best five features selected by classical methods were used, Logistic Regression and SVM achieved the highest accuracy of 96.49%. However, when quantum feature transformation was applied, the model’s accuracy reached 94.74%. Notably, training the model with all features resulted in the highest accuracy of 98.25%, surpassing the performances reported in previous studies. These findings highlight the effectiveness of both classical and quantum feature selection methods while demonstrating that using all available features yields superior classification accuracy.

The study's findings reveal that quantum machine learning approaches offer potential advantages, especially in data transformation and feature selection processes. Using quantum feature maps has shown that data can be transformed into a high-dimensional space and that this transformation can positively affect classification performance. This emphasizes the potential of quantum computing in data analysis and the importance of its integration with traditional methods.

However, the current limitations of quantum computing and high computational costs are challenges in practical applications. The development and optimization of quantum computers are critical for the more efficient and economical application of these methods on large-scale datasets. The high accuracy rates obtained with classical methods show the effectiveness of existing methods and the classification success in the dataset. Although quantum feature maps have been observed to be superior to classical methods in some cases, more research is needed to determine how effective quantum methods will be for each data set and problem.

5.3. Suggestions

1. Improving Quantum Computing: Increasing the computational capacity of quantum computers and optimizing algorithms is essential for the effective use of quantum machine learning methods on larger

data sets and real-world applications.

2. Testing with Different Data Sets: Testing quantum feature maps and quantum machine learning methods on different data sets and application domains will be useful to evaluate these methods' overall performance and validity.

3. Investigation of Hybrid Models: Combinations of quantum and classical machine learning methods, especially the development of hybrid models, can provide higher performance and flexibility by leveraging the advantages of both approaches.

4. More in-depth study of quantum machine learning in applied research, for example, in areas such as health data, financial analyses, and big data analysis, can demonstrate the benefits of quantum technology in practical applications.

5.4. Future Studies

Future work should include a more comprehensive evaluation of quantum machine learning methods and testing them on various data sets. Furthermore, overcoming the limitations of quantum computing and addressing the challenges faced in practical applications will allow us to better evaluate the potential of quantum technologies. Studies investigating the integration of quantum and classical approaches can increase the knowledge in this field and provide more effective and efficient solutions.

For this study, in the future, different feature selection techniques will be tested in both classical and quantum machine learning, and comparisons will continue using different algorithm models and performance metrics.

6 Acknowledgements

Part of this work was presented orally at the 6th International Conference on Data Science and Applications 2024 (ICONDATA'24).

References

[1] O. Ginsburg *et al.*, "Breast cancer early detection: A phased approach to implementation," *Cancer*, vol. 126, no. S10, pp. 2379–2393, 2020, doi: 10.1002/CNCR.32887.

[2] Emily Grumbling and Mark Horowitz, "Quantum Computing: Progress and Prospects." Accessed: Oct. 04, 2024. [Online]. Available: https://books.google.com.tr/books?hl=en&lr=&id=jjiPDwAAQBAJ&oi=fnd&pg=PR1&dq=Quantum+computers+leverage+the+principles+of+quantum+mechanics+to+perform+computations+that+differ+fundamentally+from+classical+computers.+Unlike+classical+bits,+which+represent+information+as+either+0+or+1,+quantum+bits+(qubits)+can+exist+in+superpositions+of+both+0+and+1+simultaneously,+enabling+quantum+computers+to+process+complex+calculations+more+efficiently.+&ots=flQcusQuaB&sig=qlo2lGDgkp5ScptYZ6lhIRReUZw&redir_esc=y#v=onepage&q&f=false

[3] J. D. Martín-Guerrero and L. Lamata, "Quantum Machine Learning: A tutorial," *Neurocomputing*, vol. 470, pp. 457–461, Jan. 2022, doi: 10.1016/J.NEUCOM.2021.02.102.

[4] P. Date, C. Schuman, R. Patton, and T. Potok, "A Classical-Quantum Hybrid Approach for Unsupervised Probabilistic Machine Learning," *Lecture Notes in Networks and Systems*, vol. 70, pp. 98–117, 2020, doi: 10.1007/978-3-030-12385-7_9.

[5] S. B. Ramezani, A. Sommers, H. K. Manchukonda, S. Rahimi, and A. Amirlatifi, "Machine Learning Algorithms in Quantum Computing: A Survey," *Proceedings of the International Joint Conference on Neural Networks*, Jul. 2020, doi: 10.1109/IJCNN48605.2020.9207714.

[6] H. Kwon, H. Lee, and J. Bae, "Feature Map for Quantum Data in Classification," *2024 International Conference on Quantum Communications, Networking, and Computing (QCNC)*, pp. 41–48, Jul. 2024, doi: 10.1109/QCNC62729.2024.00016.

[7] J. Maroco, D. Silva, A. Rodrigues, M. Guerreiro, I. Santana, and A. De Mendonça, "Data mining methods in the prediction of Dementia: A real-data comparison of the accuracy, sensitivity and specificity of linear discriminant analysis, logistic regression, neural networks, support vector machines, classification trees and random forests," *BMC Res Notes*, vol. 4, no. 1, pp. 1–14, Aug. 2011, doi: 10.1186/1756-0500-4-299/FIGURES/8.

[8] H. Kwon, H. Lee, and J. Bae, "Feature Map for Quantum Data in Classification," *Proceedings - 2024 International Conference on Quantum Communications, Networking, and Computing, QCNC 2024*, pp. 41–48, 2024, doi: 10.1109/QCNC62729.2024.00016.

[9] J. B. Prajapati, H. Paliwal, B. G. Prajapati, S. Saikia, and R. Pandey, "Quantum Machine Learning in Prediction of Breast Cancer," *Studies in Computational Intelligence*, vol. 1085, pp. 351–382, 2023, doi: 10.1007/978-981-19-9530-9_19.

[10] H. Wang, "A novel feature selection method based on quantum support vector machine," *Phys Scr*, vol. 99, no. 5, p. 056006, Apr. 2024, doi: 10.1088/1402-4896/AD36EF.

[11] H. Patel, S. Kamthekar, D. Prajapati, and R. Agarwal, "Quantum Inspired Image Classification: A Hybrid SVM Framework," *2024 International Conference on Emerging Smart Computing and Informatics, ESCI 2024*, 2024, doi: 10.1109/ESCI59607.2024.10497230.

[12] P. Patil, M. Sharma, R. Rewatkar, and B. Fulkar, "Detecting Breast Cancer: A Comparative Study of Various Machine Learning Models," *2024 Parul International Conference on Engineering and Technology, PICET 2024*, 2024, doi: 10.1109/PICET60765.2024.10716141.

[13] J. A. M. Sidey-Gibbons and C. J. Sidey-Gibbons, "Machine learning in medicine: a practical introduction," *BMC Med Res Methodol*, vol. 19, no. 1, pp. 1–18, Mar. 2019, doi: 10.1186/S12874-019-0681-4/TABLES/5.

[14] A. Sharma, S. Kulshrestha, and S. B Daniel, "Machine Learning Approaches for Cancer Detection," *International Journal of Engineering and Manufacturing*, vol. 8, no. 2, pp. 45–55, Mar. 2018, doi: 10.5815/IJEM.2018.02.05.

[15] "Cirq | Google Quantum AI." Accessed: Sep. 06, 2024. [Online]. Available: https://quantumai.google/cirq

[16] C. P. Williams, "Quantum Gates," pp. 51–122, 2011, doi: 10.1007/978-1-84628-887-6_2.

[17] "colab.google." Accessed: Oct. 04, 2024. [Online]. Available: https://colab.google/

[18] "NumPy -." Accessed: Oct. 04, 2024. [Online]. Available: https://numpy.org/

[19] "Matplotlib — Visualization with Python." Accessed: Oct. 04, 2024. [Online]. Available: https://matplotlib.org/

[20] "load_breast_cancer — scikit-learn 1.5.1 documentation." Accessed: Sep. 06, 2024. [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.datasets.load_breast_cancer.html

[21] "Breast Cancer Wisconsin (Diagnostic) - UCI Machine Learning Repository." Accessed: Feb. 01, 2025. [Online]. Available: https://archive.ics.uci.edu/dataset/17/breast+cancer+wisconsin+diagnostic

[22] A. A. A. Aayush, J. Sundaram, S. Devaraju, S. Jayaprakash, H. Anandaram, and C. Manivasagan, "Diabetic disease prediction using machine learning models and algorithms for early classification and diagnosis assessment," *Machine Learning and Deep Learning Techniques for Medical Image Recognition*, pp. 217–244, Dec. 2023, doi: 10.1201/9781003366249-13/DIABETIC-DISEASE-PREDICTION-USING-MACHINE-LEARNING-MODELS-ALGORITHMS-EARLY-CLASSIFICATION-DIAGNOSIS-ASSESSMENT-AAYUSH-JAWAHAR-SUNDARAM-DEVARAJU-SUJITH-JAYAPRAKASH-HARISHCHANDER-ANANDARAM-MANIVASAGAN.

[23] S. Raschka, "Model Evaluation, Model Selection, and Algorithm Selection in Machine Learning," Nov. 2018, Accessed: Feb. 01, 2025. [Online]. Available: https://arxiv.org/abs/1811.12808v3

[24] "SVC — scikit-learn 1.5.1 documentation." Accessed: Sep. 06, 2024. [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html

[25] "DecisionTreeClassifier — scikit-learn 1.5.1 documentation." Accessed: Sep. 06, 2024. [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html

[26] "RandomForestClassifier — scikit-learn 1.5.1 documentation." Accessed: Sep. 06, 2024. [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html

[27] "LogisticRegression — scikit-learn 1.5.1 documentation." Accessed: Sep. 06, 2024. [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html

[28] D. J. Shepherd, "On the role of Hadamard Gates in quantum circuits," *Quantum Inf Process*, vol. 5, no. 3, pp. 161–177, Jun. 2006, doi: 10.1007/S11128-006-0023-4/METRICS.

[29] D. M. Zajac *et al.*, "Resonantly driven CNOT gate for electron spins," *Science (1979)*, vol. 359, no. 6374, pp. 439–442, Jan. 2018, doi: 10.1126/SCIENCE.AAO5965/SUPPL_FILE/AAO5965_ZAJAC_SM.PDF.

[30] R. Horodecki, P. Horodecki, M. Horodecki, and K. Horodecki, "Quantum entanglement," *Rev Mod Phys*, vol. 81, no. 2, pp. 865–942, Jun. 2009, doi: 10.1103/REVMODPHYS.81.865/FIGURES/3/MEDIUM.

[31] J. R. Friedman, V. Patel, W. Chen, S. K. Tolpygo, and J. E. Lukens, "Quantum superposition of distinct macroscopic states," *Nature 2000 406:6791*, vol. 406, no. 6791, pp. 43–46, Jul. 2000, doi: 10.1038/35017505.

[32] "Quantum Feature Map — PennyLane." Accessed: Sep. 06, 2024. [Online]. Available: https://pennylane.ai/qml/glossary/quantum_feature_map/

[33] M. Yin, J. W. Vaughan, and H. Wallach, "Understanding the effect of accuracy on trust in machine learning models," *Conference on Human Factors in Computing Systems - Proceedings*, May 2019, doi: 10.1145/3290605.3300509.