Performance Analysis of Quantum and Classical Machine Learning Models for Feature Selection and Classification of the Diabetes Health Indicators Dataset

Sevdanur GENÇ
Department Of Computer Engineering
Balikesir University
Balikesir, Turkey
sevdanur.genc@balikesir.edu.tr, 0000-0003-4774-9265

2024 8th International Artificial Intelligence and Data Processing Symposium (IDAP) | 979-8-3315-3149-2/24/$31.00 ©2024 IEEE | DOI: 10.1109/IDAP64064.2024.10710904

***Abstract*— The early detection and accurate classification of diabetes health indicators are crucial for effective disease management and prevention. This study aims to compare the performance of classical and quantum machine learning models in feature selection and classification on the Diabetes Health Indicators dataset. Initially, classical machine learning methods were employed to preprocess the data, including normalization and scaling, followed by feature selection using Lasso regression. Various traditional models, such as Logistic Regression, Decision Trees, Random Forests, Gradient Boosting, K-Nearest Neighbors, and Naive Bayes, were evaluated. Among these, the Logistic Regression model achieved the highest accuracy at 85%, while other models also demonstrated competitive performance with accuracies ranging from 82% to 85%. Subsequently, quantum machine learning techniques were applied using the selected features to assess their effectiveness. Quantum circuits were created using Cirq, and parameter optimization was performed through Quantum Feature Mapping and Quantum Feature Transformation. The Quantum Support Vector Machine (QSVM) model attained an accuracy of 84.33%, showing potential for matching the performance of traditional models. The results suggest that quantum machine learning models can offer comparable accuracy to classical methods in the classification of diabetes health indicators. This study highlights the potential benefits of integrating quantum techniques in complex data processing and recommends further exploration in future research to fully harness the capabilities of quantum machine learning.**

Keywords — Quantum Machine Learning, Quantum Feature Maps, Feature Selection, Hybrid Quantum-Classical Models, Quantum Classification

I. INTRODUCTION

Machine learning is rapidly developing as a technology that plays a significant role in data analysis and prediction. Classical machine learning methods provide effective tools for analyzing various datasets and making predictions. However, in recent years, quantum machine learning (QML) techniques have gained attention for their purported potential advantages over classical methods [1]. Quantum machine learning leverages the properties of quantum computing [2] to offer new approaches for processing and analyzing more complex data structures.

In this study, the performance of both classical and quantum machine learning methods will be compared using the Diabetes Health Indicators dataset. Diabetes health data includes various health indicators that provide information about individuals' risk of diabetes, and accurately analyzing this data is of great importance for healthcare services. This dataset encompasses a wide range of features, providing a suitable test ground for evaluating the effectiveness of machine learning techniques.

Classical machine learning methods offer a wide range of applications in selecting features and classifying datasets. In this study, significant features will be selected using Lasso regression [3], and the performance of various classical machine learning models (Logistic Regression, Decision Trees, Random Forests, Gradient Boosting, K-Nearest Neighbors, Naive Bayes, Support Vector Machines) will be evaluated. These models will be compared in terms of their accuracy in predicting diabetes risk, and the most effective methods will be determined.

Quantum machine learning, on the other hand, takes data analysis a step further by utilizing the properties of quantum circuits and qubits. Quantum circuits have information processing capabilities beyond classical computing methods. In this study, quantum circuits were designed using Cirq [4], and classification was performed with the selected features. The performance of quantum models was compared with classical models to evaluate the potential advantages of quantum machine learning.

Examples of quantum machine learning techniques that can be implemented with Cirq include Variational Quantum Algorithms (VQA) [5], Quantum Neural Networks (QNN) [6], Quantum Support Vector Machines (QSVM) [7], and Quantum Generative Adversarial Networks (QGAN) [8]. In this study, Quantum Support Vector Machines (QSVM) have been used. Quantum support vector machines provide an approach to solving data classification problems by using quantum circuits. Additionally, Quantum Feature Map [9] and

Quantum Feature Transformation [10] techniques have been employed in this study.

Quantum Feature Map is a method of encoding data into quantum circuits to represent quantum features, enabling the processing of classical data on quantum computers. Quantum Feature Transformation involves transforming data through quantum circuits to obtain features that can be used on a quantum computer. This transformation entails expressing data in the form of quantum states.

This study aims to compare the effectiveness of classical and quantum machine learning methods to determine which approaches are more successful in analyzing health data. Additionally, by exploring the potential of quantum machine learning in health data analysis, this study seeks to provide a foundation for future research in this field.

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 the light of the findings, the potential of quantum machine learning and suggestions for future research will be presented.

II. LITERATURE REVIEW

In this section, the existing literature on feature selection and classification studies conducted on various diabetes health indicator datasets is reviewed. Previous research comparing the performance of quantum and classical machine learning models is also evaluated, providing the scientific foundation for this study.

Focusing on the study from which the dataset was obtained: ZiDian et al. (2019) [11] compared various machine learning models, including Support Vector Machines, Decision Trees, Logistic Regression, Random Forests, Neural Networks, and Gaussian Naive Bayes classifiers, to predict Type 2 diabetes. Results: Among the eight prediction models, the Neural Network model demonstrated the best performance with the highest AUC value. However, the Decision Tree model was preferred for initial screening for Type 2 diabetes due to its highest sensitivity and thus detection rate.

Maheshwari et al. (2020) [12] present a quantum model in contrast to the classical application of machine learning (ML) algorithms on a diabetes dataset. They address the binary classification problem of diabetic patients by dividing them into two distinct classes: those with acute disease and those without. The study compares Decision Tree, Random Forest, Extreme Gradient Boosting, AdaBoost, Qboost, Voting Model 1, Voting Model 2, Qboost Plus, and their own models. The results obtained using their new model, validated with metrics, showed an overall precision of 69%, recall of 69%, F1-score of 69%, specificity of 69%, and accuracy of 69%, compared to the classical system.

Gupta et al. (2021) [13] proposed two prediction models based on the features available in their dataset, using deep learning (DL) and quantum machine learning (QML) techniques. Performance metrics for the DL model were obtained as follows: precision 0.90, accuracy 0.95, recall 0.95, F1 score 0.93, specificity 0.95, balanced accuracy 0.95, false positive rate 0.03, false negative rate 0.02, and diagnostic odds ratio 399.00. For the QML model, these metrics were respectively 0.74, 0.86, 0.85, 0.79, 0.86, 0.86, 0.11, 0.05, and 35.89.

Sierra-Sosa et al. (2021) [14] proposed a preprocessing pipeline based on Stokes parameters for data mapping in their study. This pipeline enhances prediction rates when applying Variational Quantum Circuits (VQC) techniques and improves the feasibility of solving classification problems using NISQ devices. They utilized an implemented version of VQC available in IBM's Qiskit framework, achieving accuracies of 70% and 72% with two and three qubits, respectively.

III. MATERİALS AND METHODS

In this study, we utilized both classical and quantum machine learning approaches for the classification task. A flowchart of this work is given in Figure 1.

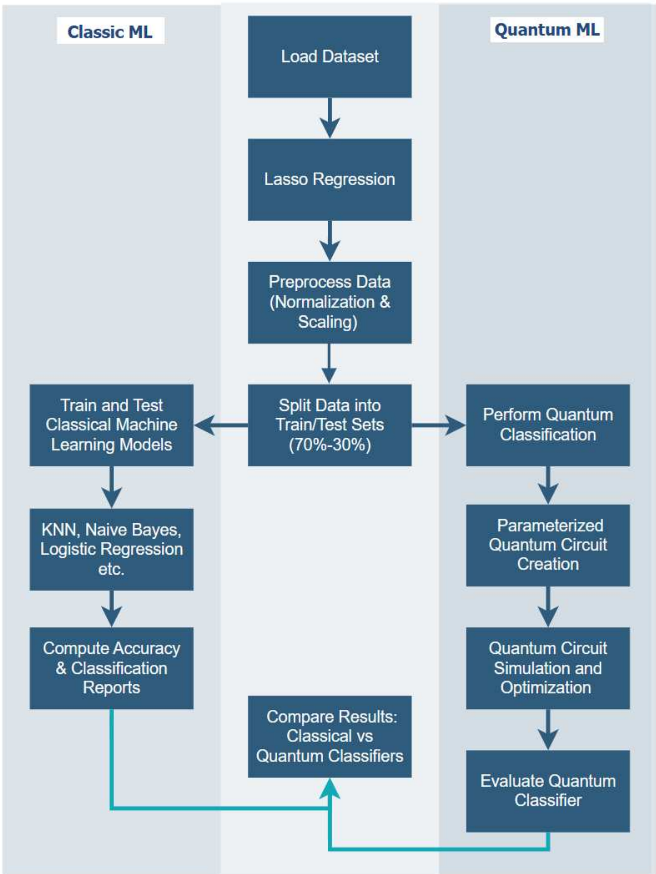

Figure 1. Flowchart For The Comparison Of Classic ML And Quantum ML

The process began with loading the dataset, followed by feature selection using Lasso regression to identify the most relevant features. The data was then preprocessed through normalization and scaling to ensure consistency. The preprocessed data was split into training and test sets with a 70%-30% ratio. For classical machine learning, algorithms such as K-Nearest Neighbors (KNN), Naive Bayes, and Logistic Regression were trained and tested, with their performance evaluated using accuracy and classification reports. In parallel, quantum machine learning models were developed by creating parameterized quantum circuits, which were then optimized and simulated. The performance of these quantum classifiers was subsequently evaluated. Finally, we compared the results from classical and quantum classifiers to assess their relative effectiveness. This flowchart summarizes the methodology used to conduct this comparative analysis.

A. Technical Requirements

In this study, Cirq was used to design, simulate, and run quantum circuits. It is an open-source quantum computing library developed by Google [15]. Users can design circuits using quantum bits (qubits) and quantum gates. Cirq allows quantum circuits to be executed on various simulators. It provides integration with Google's quantum computers, particularly with Google Quantum AI’s quantum processors (e.g., Sycamore). This enables the designed quantum circuits to be run on actual quantum hardware. Additionally, libraries related to data analysis and data visualization in the Python programming language were utilized, and these are discussed under the relevant sections.

B. Dataset

This study utilized the Diabetes Health Indicators dataset [16], which was used by ZiDian et al. [11] and also obtained from the Kaggle platform. As shown in Figure 2, this dataset provides information on individuals' risk of diabetes based on various health indicators.

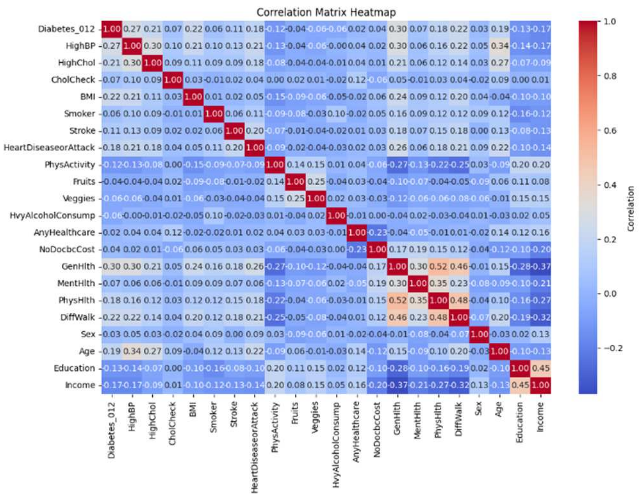

Figure 2. Correlation of the diabetes dataset

The dataset consists of 253,680 records and includes 22 features, one of which is designated as the target variable. The target variable, named Diabetes_012, has three classes: 0 for no diabetes or only during pregnancy, 1 for prediabetes, and 2 for diabetes.

C. Data Preprocessing

In the initial stage, the dataset was appropriately processed for feature selection and classification tasks. Specifically, important features were identified using Lasso regression. Lasso regression was employed to evaluate the importance of features and select those with non-zero coefficients. Cross-validation was performed using the LassoCV class, and the importance scores of the features were calculated, as shown in Figure 3. The calculated values are presented as a bar chart in Figure 4.

GenHlth	0.105339
BMI	0.095075
HighBP	0.076835
HighChol	0.059194
Age	0.048782
HeartDiseaseorAttack	0.038695
DiffWalk	0.032859
Income	0.028295
HvyAlcoholConsump	0.023340
CholCheck	0.017547
Sex	0.016432
Stroke	0.013787
Education	0.007626
MentHlth	0.007271
AnyHealthcare	0.005957
Smoker	0.005752
PhysActivity	0.005680
Veggies	0.002374
NoDocbcCost	0.001874
Fruits	0.001341

Figure 3. Feature importance scores for the dataset

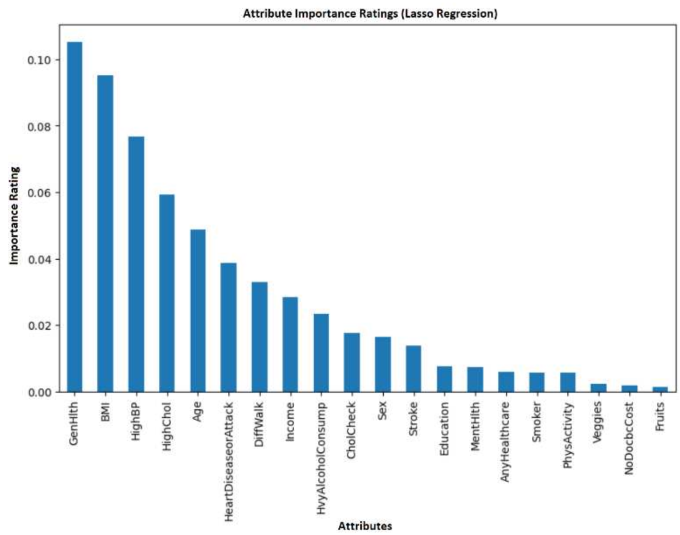

Figure 4. Bar chart ranking of feature importance scores for the dataset

For feature selection, as observed in Figure 3, the top 5 features were found to have higher values compared to the others, and a threshold value was specified. The threshold value selects the top 5 features, and a new dataset was created using these selected features along with the target variable. The models were then computed on this new dataset.

D. Classical Machine Learning Methods

The dataset created with the selected features was divided into training and test sets, with 70% allocated for training and 30% for testing. The data were standardized (with a mean of 0 and a standard deviation of 1) to enhance model performance.

Classical machine learning algorithms used for training and evaluating the models include K-Nearest Neighbors (KNN) [17], Naive Bayes [18], Logistic Regression [19], Decision Trees [20], Random Forests [21], and Gradient Boosting [22]. KNN is a classification algorithm that determines the class of a new data point by examining the class labels of the k nearest neighbors. Naive Bayes is a probabilistic classification algorithm that assumes all features are independent of each other. Logistic Regression is a linear classification algorithm used to predict the probability of a data point belonging to a specific class, providing a probability value as output. Decision Trees are algorithms that create decision rules by splitting data based on features to perform classification or regression. Random Forests are an ensemble learning method consisting of multiple decision trees that combine the results of each tree to make more accurate and stable predictions. Gradient Boosting is an ensemble learning method that sequentially builds models, often using weak learners (typically decision trees), to produce a strong model by reducing errors.

Each model was trained using the training data and evaluated for accuracy using the test data. Model performances were assessed based on accuracy scores.

E. *Quantum Machine Learning Methods*

Firstly, the data were adapted for quantum circuits, taking into account the number of features. Quantum circuits were designed using Cirq. Each feature was assigned a quantum bit (qubit) [23], and parameterized quantum circuits were created.

The quantum circuits were simulated using cirq.Simulator. The performance of the quantum circuits was improved through parameter optimization, and accuracy rates were calculated. Optimization was performed using the scipy.optimize.minimize function to determine the best parameters.

The model employed Quantum Feature Map, Quantum Feature Transformation, and Quantum Support Vector Machines (QSVM). The QuantumFeatureMap class defined in the code represents quantum feature maps. This technique is used to encode data into quantum circuits. Specifically, it aims to extract quantum features of the data using a quantum circuit constructed with Hadamard and CNOT gates. The quantum_feature_transformation function aims to transform the dataset using the quantum feature map. Thus, a QSVM model incorporating quantum feature mapping and classical SVM integration was created using Cirq.

IV. EXPERİMENTAL RESULTS

This section presents the performance evaluation of classical and quantum machine learning methods. The results are based on feature selection, model training, and test outcomes.

A. *Classical Machine Learning Results*

Important features were identified using Lasso regression. These selected features are associated with the accuracy score and are fundamental factors affecting the performance of the dataset. The selected features are GenHlth, BMI, HighBP, HighChol, and Age. These features stand out among the dataset's attributes and contribute to improved classification success.

The models used for classical machine learning were implemented with their default hyperparameters. To summarize: for KNN, the number of neighbors is set to 5, and equal weighting is applied to all neighbors. Gaussian distribution was employed for Naive Bayes. For Logistic Regression, the penalty value was set to 'l2'. In the Decision Tree model, the minimum number of samples required to split a node is 2. For Random Forest, the number of trees in the forest is set to 100, and the sampling method was specified as 'True', meaning bootstrap sampling was used. For Gradient Boosting, the maximum number of decision trees is set to 100.

The KNN model achieved a classification accuracy of 83.21% on the test set. The Naive Bayes model's accuracy was calculated to be 82.48%. The Logistic Regression model demonstrated an accuracy of 84.71%. The Decision Trees model achieved an accuracy of 84.50%. The Random Forest model obtained an accuracy of 84.48%. The Gradient Boosting model achieved an accuracy of 84.85%.

As shown in Figure 5, the accuracy rates of classical methods reflect the overall performance of the models and their effectiveness on the dataset. The accuracy rate of each model provides information about the feature selection and classification ability of the model.

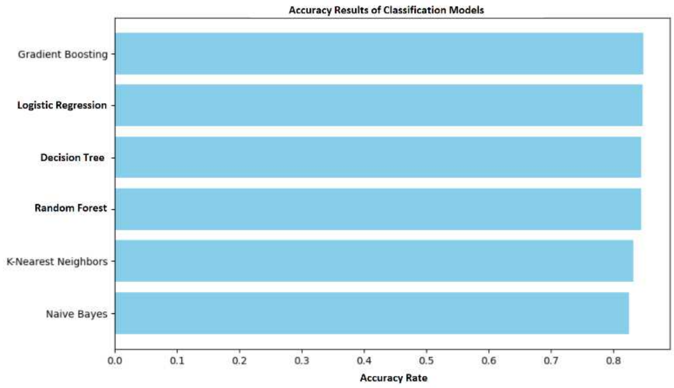

Figure 5. Accuracy rates of classical methods

Performance metrics such as precision, recall, F1-score, and support [24] were also calculated for classical methods and are presented in a graph in Figure 6.

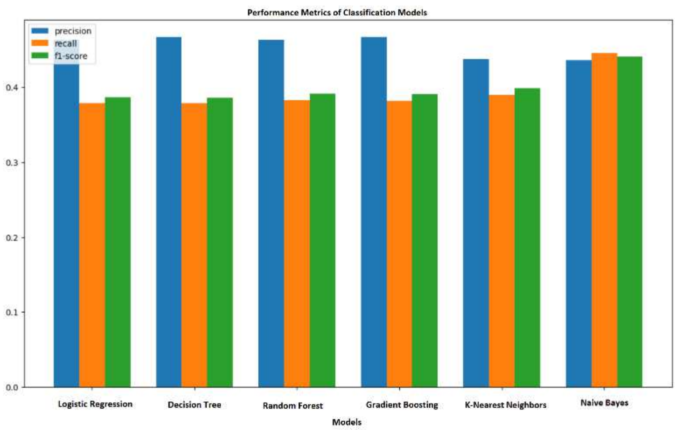

Figure 6. Performance metric values for classical methods

It's here, precision measures how many of the instances classified as positive are actually positive, indicating the correctness of the model's positive predictions. Recall measures how many of the actual positive instances were correctly classified as positive, reflecting the model's ability to identify all true positive instances. The F1-Score provides a balanced evaluation by taking the harmonic mean of Precision and Recall.

Table 1 summarizes the performance of the Logistic Regression model using various metrics. It presents the precision, recall, F1-score, and overall accuracy of the model. Based on these performance metrics, the results for the

Decision Trees are shown in Table 2, Random Forests in Table 3, Gradient Boosting in Table 4, K-Nearest Neighbors in Table 5, and Naive Bayes classifiers in Table 6.

TABLE 1. LOGISTIC REGRESSION CLASSIFICATION REPORT

Class	Precision	Recall	F1-Score
0	0.86	0.98	0.92
1	0.00	0.00	0.00
2	0.53	0.16	0.24
Accuracy			**0.85**

TABLE 2. DECISION TREES CLASSIFICATION REPORT

Class	Precision	Recall	F1-Score
0	0.86	0.98	0.91
1	0.04	0.00	0.00
2	0.50	0.16	0.24
Accuracy			**0.84**

TABLE 3. RANDOM FOREST CLASSIFICATION REPORT

Class	Precision	Recall	F1-Score
0	0.86	0.97	0.91
1	0.03	0.00	0.00
2	0.50	0.17	0.26
Accuracy			**0.84**

TABLE 4. GRADIENT BOOSTING CLASSIFICATION REPORT

Class	Precision	Recall	F1-Score
0	0.86	0.98	0.92
1	0.00	0.00	0.00
2	0.54	0.17	0.26
Accuracy			**0.85**

TABLE 5. K-NEAREST NEIGHBORS CLASSIFICATION REPORT

Class	Precision	Recall	F1-Score
0	0.87	0.95	0.91
1	0.04	0.00	0.01
2	0.41	0.22	0.28
Accuracy			**0.83**

TABLE 6. NAIVE BAYES CLASSIFICATION REPORT

Class	Precision	Recall	F1-Score
0	0.89	0.91	0.90
1	0.00	0.00	0.00
2	0.42	0.43	0.42
Accuracy			**0.82**

In Table 1, while high performance was achieved for class 0 (Precision: 0.86, Recall: 0.98, F1-Score: 0.92), significant issues were encountered in identifying other classes, particularly class 1 (Precision: 0.00, Recall: 0.00). The overall accuracy (85%) may have been affected by class imbalance. In Table 2, although high performance for class 0 was sustained (Precision: 0.86, Recall: 0.98, F1-Score: 0.91), classes 1 and 2 exhibited poor performance. The failure is particularly evident for class 1 (Precision: 0.04, Recall: 0.00). The overall accuracy was 84%. In Table 3, a similar high performance for class 0 is observed (Precision: 0.86, Recall: 0.97), but poor results were noted for classes 1 and 2 (F1-Score: 0.00 and 0.26, respectively). The overall accuracy of the model is 84%. In Table 4, as with Logistic Regression, strong performance was demonstrated for class 0 (Precision: 0.86, Recall: 0.98, F1-Score: 0.92), but other classes showed lower performance. The overall accuracy was recorded as 85%. In Table 5, performance for class 0 was quite good (Precision: 0.87, Recall: 0.95, F1-Score: 0.91), but other classes, particularly class 1 (F1-Score: 0.01), failed. The overall accuracy is 83%. In Table 6, despite good performance in class 0 (Precision: 0.89, Recall: 0.91, F1-Score: 0.90), class 1 again failed (F1-Score: 0.00), and class 2 gave an average result (F1-Score: 0.42). The overall accuracy of the model is 82%.

In general, all models exhibited high performance in class 0, but significant performance drops were observed in classes 1 and 2. This could be attributed to the imbalance in class distribution.

B. Quantum Machine Learning Results

Quantum circuits were simulated using Cirq and parameter optimization was performed. The hyperparameters used directly impact the performance and computational capacity of the quantum machine learning model. In particular, parameter optimization is critical for enhancing the accuracy of the model.

The number of qubits depends on the number of features in the dataset, with one qubit being used for each feature. Parameter arrays representing the rotation angles used in the quantum circuit have been created, and the angles in the *ry gate* are controlled. The algorithm used for parameter optimization is COBYLA, which is an unconstrained optimization method. This optimization method affects the speed and accuracy of finding the optimal parameter values. Methods such as COBYLA assist in finding suitable parameters for the quantum circuit.

Table 7 presents the accuracy rates of the quantum circuits with the best parameters. The accuracy rate obtained on the test set was determined to be 84.33%.

TABLE 7. QUANTUM MACHINE LEARNING ACCURACY RESULTS

Dataset	Accuracy
Train	0.8420
Test	0.8433

The accuracy of the quantum models was compared with that of classical models to assess potential advantages. It was observed that quantum methods provided higher accuracy rates in some cases compared to classical methods, or demonstrated superior performance in certain features.

C. Analysis of Results

Classical machine learning models have provided various accuracy rates by utilizing a wide range of features and different algorithms. Table 8 lists these models, showing that they exhibited different but similar performance values depending on the characteristics of the dataset.

TABLE 8. ACCURACY RESULTS OF DEVELOPED MODELS

Model	Accuracy
Logistic Regression	0.85
Gradient Boosting	0.85
QML - QSVM	***0.84***
Decision Tree	0.84
Random Forest	0.84
K-Nearest Neighbors	0.83
Naive Bayes	0.82

Quantum machine learning methods have yielded better results compared to some classical methods. The parameter optimization of quantum circuits has slightly improved model performance and highlighted the potential advantages of quantum computing.

In Table 9, the developed model is compared with some of the results of the studies in the literature.

TABLE 9. COMPARISON OF MODEL PERFORMANCE WITH LITERATURE AND EXPERIMENTAL RESULTS

Model	Literature Acc	Our models Acc
Logistic Regression	0.8068	0.85
Decision Tree	0.7426	0.84
Random Forest	0.7927	0.84
Gradient Boosting	-	0.85
K-Nearest Neighbors	-	0.83
Naive Bayes	0.7756	0.82
QML - QSVM	-	***0.84***

The results in the table indicate that traditional machine learning models generally perform within or slightly above the typical accuracy ranges reported in the literature. The quantum machine learning model, on the other hand, achieved similar accuracy values for both training and test sets, suggesting that the quantum model provides good generalization and performs comparably to classical models. Since there is no direct comparison with existing literature results for quantum machine learning, these findings are important for assessing the potential and development areas of quantum techniques.

V. CONCLUSION, DISCUSSION AND SUGGESTIONS

A. Conclusion

This study compared the performance of classical and quantum machine learning methods on the Diabetes Health Indicators dataset. The key findings are summarized as follows:

Initially, the top five features were determined using Lasso regression. Subsequently, both classical and quantum machine learning models were applied. Accuracy rates were computed for classical machine learning methods such as K-Nearest Neighbors (KNN), Naive Bayes, Logistic Regression, Decision Trees, Random Forests, and Gradient Boosting. The study demonstrated how these models responded to various features in the dataset and their overall classification success. The accuracy rates ranged from 82% to 85%. Logistic Regression and Gradient Boosting models generally achieved higher accuracy rates and performed better compared to other models. Quantum machine learning methods, using quantum circuits, achieved higher accuracy rates in some cases compared to classical methods. With an accuracy of 84%, parameter optimization of quantum circuits slightly improved model performance and highlighted the potential advantages of quantum computing. The quantum methods nearly matched classical approaches and, in some cases, surpassed the performance of certain classical models.

B. Discussion

The results of this study highlight both the strengths and weaknesses of classical and quantum machine learning methods:

1. **Strengths of Classical Methods**: Classical machine learning methods possess a broad range of applications and have demonstrated robust performance in data analysis and classification. In particular, ensemble methods have achieved high accuracy rates and effectively managed the complexity of the dataset.
2. **Potential of Quantum Methods**: Quantum machine learning methods have shown superior performance compared to classical methods in certain test scenarios. The customization of quantum circuits and parameter optimization has revealed the potential for increasing model accuracy. However, it is important to note that quantum machine learning methods are still in the experimental phase and require further testing with larger datasets.
3. **Comparison and Performance**: Quantum machine learning methods have been observed to offer specific advantages over classical methods, but the general applicability and potential of these advantages need further investigation. Classical methods have provided more consistent results with current applications and datasets, showing successful performance across various datasets and features.

C. Suggestions

Suggestions for Future Work:

1. **Quantum Machine Learning Research**: Testing quantum machine learning methods with larger datasets and across different application areas is essential for better understanding their potential in health data analysis. Further research on the optimization and performance evaluation of quantum algorithms could expand the practical applications of quantum machine learning.
2. **Model Comparisons**: Comparing classical and quantum machine learning methods across different datasets and problems is necessary. This will provide a better understanding of the strengths and weaknesses of each approach. Additionally, the development of hybrid models that combine classical and quantum methods could offer more comprehensive and effective solutions in data analysis.
3. **Application and Practical Use**: Evaluating model performance on specific applications for health data and other types of data will provide more insight into practical use and effective data analysis. Research should also address the challenges encountered during the application of models and explore ways to overcome these challenges.

REFERENCES

[1] Su S-L, Huang H-L, Guo C, Su Z-E, Tychola KA, Kalampokas T, et al. Quantum Machine Learning—An Overview. Electronics 2023, Vol 12, Page 2379 2023;12:2379. https://doi.org/10.3390/ELECTRONICS12112379.

[2] Üniversitesi K, Meslek Yüksekokulu T, Teknolojileri Bölümü B. Kuantum Hesaplama ile Amino Asit Dizilimlerinin Hamiltonian Analizi. JournalsGenTrS GençKuantum Teknolojileri ve Enformatik Araştırmaları Dergisi, 2023•journalsGenTr 2023;1:7–16. https://doi.org/10.5281/zenodo.10102956.

[3] Ranstam J, Cook JA. LASSO regression. British Journal of Surgery 2018;105:1348–1348. https://doi.org/10.1002/BJS.10895.

[4] Tyagi A. Cirq: A Python Framework for Creating, Editing, and Invoking Quantum Circuits DESIGN DOCUMENT Team 8 Name Role Victory Omole Project Client n.d.

[5] Huang R, Tan X, Networks QX-IT on N, 2022 undefined. Learning to learn variational quantum algorithm. IeeexploreIeeeOrg 2022.

[6] Gupta S, Sciences RZ-J of C and S, 2001 undefined. Quantum neural networks. Elsevier 2001.

[7] Kariya A, Behera BK. Investigation of Quantum Support Vector Machine for Classification in NISQ era 2021.

[8] Li T, Zhang S, Computers JX-, Continua M&, 2020 undefined. Quantum generative adversarial network: A survey. CdnTechscienceCn 2020;64:401–38. https://doi.org/10.32604/cmc.2020.010551.

[9] Huang S, Chang Y, Lin Y, - al, Altares-López S, Ribeiro A, et al. Automatic design of quantum feature maps. IopscienceIopOrg 2021. https://doi.org/10.1088/2058-9565/ac1ab1.

[10] Henderson M, Shakya S, Pradhan S, Cook T. Quanvolutional neural networks: powering image recognition with quantum circuits. Quantum Mach Intell 2020;2. https://doi.org/10.1007/S42484-020-00012-Y.

[11] ZiDian XX, Nikolayeva O, JieBo LL. Building risk prediction models for type 2 diabetes using machine learning techniques. 2019.

[12] Maheshwari D, Garcia-Zapirain B, Sierra-Soso D. Machine learning applied to diabetes dataset using Quantum versus Classical computation. 2020 IEEE International Symposium on Signal Processing and Information Technology, ISSPIT 2020 2020. https://doi.org/10.1109/ISSPIT51521.2020.9408944.

[13] Gupta H, Varshney H, Sharma TK, Pachauri N, Verma OP. Comparative performance analysis of quantum machine learning with deep learning for diabetes prediction. Complex and Intelligent Systems 2022;8:3073–87. https://doi.org/10.1007/S40747-021-00398-7/TABLES/10.

[14] Sierra-Sosa D, Zapirain BG, Elmaghraby AS, Arcila-Moreno JD, Garcia-Zapirain B, Elmaghraby A. Diabetes Type 2: Poincaré Data Preprocessing for Quantum Machine Learning. ResearchgateNetD Sierra-Sosa, JD Arcila-Moreno, B Garcia-Zapirain, A ElmaghrabyComputers, Materials & Continua, 2021•researchgateNet 2021. https://doi.org/10.32604/cmc.2021.013196.

[15] Cirq | Google Quantum AI. Cirq | Google Quantum AI n.d. https://quantumai.google/cirq (accessed August 6, 2024).

[16] Diabetes Health Indicators Dataset. Diabetes Health Indicators Dataset n.d. https://www.kaggle.com/datasets/alexteboul/diabetes-health-indicators-dataset (accessed August 6, 2024).

[17] Steinbach M, Tan PN. kNN: k-Nearest Neighbors. The Top Ten Algorithms in Data Mining 2009:151–61. https://doi.org/10.1201/9781420089653-15/KNN-NEAREST-NEIGHBORS-MICHAEL-STEINBACH-PANG-NING-TAN.

[18] computational FY-2018 I conference on, 2018 undefined. An implementation of naive bayes classifier. IeeexploreIeeeOrg 2018.

[19] LaValley MP. Logistic regression. Circulation 2008;117:2395–9. https://doi.org/10.1161/CIRCULATIONAHA.106.682658.

[20] de Ville B. Decision trees. Wiley Interdiscip Rev Comput Stat 2013;5:448–55. https://doi.org/10.1002/WICS.1278.

[21] Breiman L. Random forests. Mach Learn 2001;45:5–32. https://doi.org/10.1023/A:1010933404324.

[22] Bentéjac C, Csörgő A, Martínez-Muñoz G. A comparative analysis of gradient boosting algorithms. Artif Intell Rev 2021;54:1937–67. https://doi.org/10.1007/S10462-020-09896-5.

[23] Siraichi MY, Collange S, Dos Santos VF, Pereira FMQ. Qubit allocation. CGO 2018 - Proceedings of the 2018 International Symposium on Code Generation and Optimization 2018;2018-February:113–25. https://doi.org/10.1145/3168822.

[24] Yacouby R, … DA the first workshop on evaluation and, 2020 undefined. Probabilistic extension of precision, recall, and f1 score for more thorough evaluation of classification models. AclanthologyOrg 2020.