Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Quantum Neural Networks for Wind Energy Forecasting: A Comparative Study of Performance and Scalability with Classical Models

Batuhan Hangun
Yıldız Technical University
Onder Eyecioglu
onder.eyecioglu@ibu.edu.tr
Bolu Abant İzzet Baysal University
Mehar Ali
Information Technology University
Oguz Altun
Yıldız Technical University
Korhan Kayisli
Gazi University

Research Article

Keywords: Quantum neural network, Quantum circuit benchmark, Simulation time complexity, Quantum computing, Wind power prediction

Posted Date: June 30th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-9475883/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

Quantum Neural Networks for Wind Energy Forecasting: A Comparative Study of Performance and Scalability with Classical Models

Batuhan Hangun[1*], Onder Eyecioglu[2*], Mehar Ali[3], Oguz Altun[1], Korhan Kayisli[4]

[1*]Department of Computer Engineering, Yildiz Technical University, Esenler, Istanbul, 34230, Türkiye.
[2]Department of Computer Engineering, Bolu Abant Izzet Baysal University, Golkoy, Bolu, 14100, Türkiye.
[3]Department of Computer Science, Information Technology University, Arfa Software Technology Park, Punjab, 54000, Türkiye.
[4]Department of Electrical and Electronics Engineering, Gazi University, Maltepe, Ankara, 06570, Türkiye.

*Corresponding author(s). E-mail(s): batuhanhangun@gmail.com; onder.eyecioglu@ibu.edu.tr;
Contributing authors: mehar.ali@itu.edu.pk; oaltun@yildiz.edu.tr; korhankayisli@gazi.edu.tr;

Abstract

Quantum Machine Learning (QML) is an emerging field at the intersection of quantum computing and machine learning that aims to enhance classical learning methods by leveraging principles such as superposition and entanglement. However, skepticism remains regarding its practical advantages, particularly due to the limitations of noisy intermediate-scale quantum (NISQ) devices. This study addresses these concerns through a systematic evaluation of Quantum Neural Networks (QNNs), quantum-inspired counterparts of Artificial Neural Networks (ANNs). We construct and analyze 12 distinct QNN configurations by combining two quantum feature maps with six entanglement strategies for ansatz design. Experiments conducted on a wind energy dataset show that QNNs can achieve competitive or superior performance compared to classical baseline models, with the best-performing configuration (QNN-3) reaching an R^2 score of up to 0.94 when forecasting wind power output using four input features. In addition, we

provide a detailed analysis of simulation time across different configurations, highlighting trade-offs between model complexity and computational cost. These findings demonstrate the potential of QNNs as viable models for predictive tasks and contribute to the ongoing evaluation of QML in real-world applications.

Keywords: Quantum neural network, Quantum circuit benchmark, Simulation time complexity, Quantum computing, Wind power prediction

1 Introduction

Quantum computing (QC) has gained significant attention in recent years due to rapid advancements in quantum hardware, including Google's state-of-the-art Willow chip [1] and Microsoft's Majorana 1 chip [2]. Despite these developments, the field remains constrained by the limitations of noisy intermediate-scale quantum (NISQ) devices, and large-scale fault-tolerant quantum computing is still expected to take several years to mature [3]. Nevertheless, these advancements continue to drive interest in identifying practical applications where quantum computing can offer measurable advantages over classical methods. One promising direction is the integration of quantum computing with artificial intelligence (AI), giving rise to the field of Quantum Machine Learning (QML).

QML aims to enhance classical machine learning (ML) by leveraging quantum mechanical principles such as superposition and entanglement. Among QML approaches, Quantum Neural Networks (QNNs) have emerged as a key paradigm for modeling complex data relationships. Prior studies have demonstrated the applicability of QML across various domains. For example, quantum kernel-based methods have shown improved performance in fraud detection tasks [4], while hybrid quantum-classical models such as autoencoders and convolutional neural networks have achieved competitive results in anomaly detection and image analysis tasks [5–7]. These works highlight the growing versatility of QML; however, they often focus on specific applications without providing systematic comparisons across different QNN configurations.

In the energy domain, QML has been explored for tasks such as fault diagnosis, grid stability assessment, and renewable energy forecasting. Hybrid quantum-classical models have been successfully applied to power system fault classification [8], while other studies have demonstrated the potential of QML in improving prediction accuracy and feature representation for complex energy datasets [9–14]. These applications indicate that QML is particularly well-suited for handling high-dimensional and nonlinear relationships commonly found in energy systems. However, despite these promising results, existing studies lack a unified framework for benchmarking QNN architectures in real-world regression tasks.

This gap is particularly evident in wind energy forecasting, where accurate prediction of power output is critical for efficient grid integration and operational planning. While classical ML methods remain the dominant approach for such tasks [15–21], increasing data complexity poses challenges in scalability and learning performance.

Table 1 Basic descriptive statistics for the wind turbine dataset.

Variable	Mean	Median	Std. Dev.	Min	Max	Range
Temperature [$°C$]	3.94	4.205	2.041	-5.29	10.00	15.29
Pressure [hPa]	1019.46	1024.745	13.054	979.79	1035.72	55.93
Direction [θ]	243.11	247.90	55.109	100.67	359.78	259.11
Velocity [m/s]	8.65	7.635	4.241	0.32	21.07	20.75
Power [kW]	666.60	376.675	716.757	2.24	2033.12	2030.88

Although QML offers potential advantages in representing complex feature interactions, there is limited empirical evidence guiding the selection of optimal QNN architectures. In particular, there is a lack of systematic studies evaluating how different QNN configurations perform across varying dataset sizes and how their computational costs scale in simulation environments.

To address this gap, this study presents a systematic benchmark of QNN architectures for a real-world wind energy regression task. Specifically, we investigate multiple QNN configurations based on variations in ansatz design and evaluate their predictive performance and computational efficiency. In addition to performance comparison with classical models, we analyze how dataset size impacts both model accuracy and simulation runtime, providing practical insights for real-world deployment.

The main contributions of this study are summarized as follows:

1. A systematic benchmark of six QNN configurations on a real-world wind energy regression task.
2. An analysis of how dataset size affects both predictive performance and simulation time.
3. Empirical evidence that QNN training time scales linearly ($\mathcal{O}(n)$) with dataset size, with total gate count as the primary driver.
4. A stability analysis framework for comparing QNN learning dynamics across different circuit configurations.

The remainder of this paper is organized as follows: Section 2 describes the dataset and data preparation, Section 3 presents the experimental design, Section 4 discusses the results, and Section 5 concludes the study and outlines future work.

2 Dataset and Preprocessing

To demonstrate the practical application of quantum computing, this study uses a real-world temporal dataset of a wind turbine's power output. This dataset, previously utilized in our successful applications of classical machine learning regression algorithms for wind power prediction [22], comprises 4464 measurements recorded at 10-minute intervals. Each data point included four input features that influenced the power output. The measured parameters include wind speed (m/s), wind direction (θ), atmospheric pressure (hPa), ambient temperature ($°C$), and generated power output (kW) of a wind turbine [23]. The basic statistical properties of the dataset are presented in Table 1. The target feature "*Power*" ranges from 2.24 to 2033.12 kW, which has the widest range of the features.

As this study aimed to conduct an experimental analysis of the dataset size and ansatz configurations, we created four subsets from the original dataset. To better observe how the results vary with increasing dataset size, instead of using all 4464 samples, we randomly generated subsets containing 1000, 2000, 3000, and 4000 samples and used these subsets for each experiment. These subset sizes were selected to provide evenly spaced increments that allow a clear analysis of scalability trends in both predictive performance and simulation time while remaining sufficiently close to the full dataset size. For each experiment, we divided the data into training (80%) and test (20%) sets, and then used a min-max scaler (1) to scale all values to the range $[0, 1]$.

$$x_{scaled} = \frac{x_{original} - x_{min}}{x_{max} - x_{min}} \tag{1}$$

3 Methodology

3.1 Quantum Machine Learning

Quantum Machine Learning (QML) integrates quantum computing with classical machine learning to exploit quantum mechanical properties such as superposition and entanglement. Depending on the nature of the data and computational paradigm, QML approaches can be categorized into four types: Classical-Classical (CC), Classical-Quantum (CQ), Quantum-Classical (QC), and Quantum-Quantum (QQ) [24]. These categories are illustrated in Figure 1.

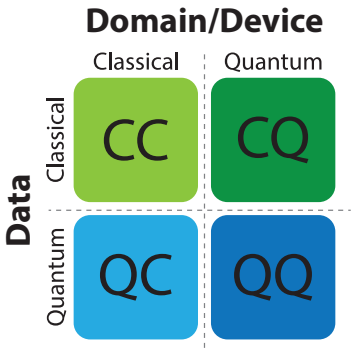


Fig. 1 Classification of QML approaches based on data type and processing environment.

In this study, we adopt the Classical-Quantum (CQ) paradigm, where classical data are processed using quantum models. Specifically, we employ Quantum Neural Networks (QNNs) for a regression task in a renewable energy context.

3.2 Quantum Neural Networks

QNNs are hybrid quantum-classical models consisting of a parameterized quantum circuit and a classical optimizer. The general architecture used in this study is shown

in Figure 2. The quantum layer encodes and processes input data, while the classical layer updates circuit parameters based on the optimization objective.

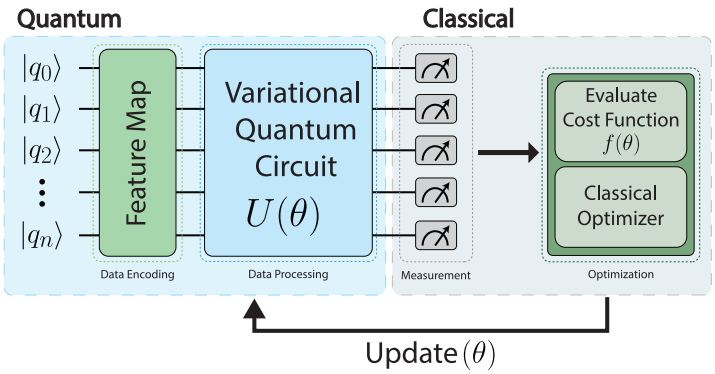


Fig. 2 General architecture of the Quantum Neural Network (QNN).

A QNN typically consists of four components: data encoding, quantum processing (ansatz), measurement, and optimization. In this work, we focus on how encoding strategies and ansatz design influence predictive performance and simulation efficiency.

3.3 Data Encoding

We employed the Z Feature Map for encoding classical input data into quantum states. This choice was motivated by our previous study [25], which demonstrated that the Z Feature Map achieved approximately three times better predictive performance than the ZZ Feature Map on the same dataset. Additionally, the Z Feature Map produces shallow circuits without entanglement, making it well-suited for both simulation and near-term quantum hardware.

The mathematical formulation of the Z Feature Map is provided in Appendix A.

3.4 Quantum Circuit Design (Ansatz)

After encoding, data are processed using a parameterized quantum circuit (ansatz). In this study, we employed the Real Amplitude ansatz, which uses rotational gates and controlled entangling operations. This ansatz is widely adopted due to its balance between expressibility and hardware efficiency [26–28].

To investigate the effect of circuit structure, we evaluated six different entanglement strategies: *Full*, *Linear*, *Circular*, *Shifted-Circular Alternating (SCA)*, *Reverse Linear*, and *Pairwise*. These configurations vary in how qubits are interconnected, which directly impacts both learning capability and computational cost.

The mathematical formulation of the ansatz and gate operations is provided in Appendix A.

3.5 QNN Configurations

We constructed six QNN models by combining the Z Feature Map with different ansatz entanglement strategies. The configurations are summarized in Table 2. Circuit diagrams for all configurations are provided in Appendix B to support reproducibility.

Table 2 QNN configurations used in this study.

QNN Configuration	Feature Map	Ansatz
QNN-1	Z Feature Map	Full
QNN-2	Z Feature Map	Linear
QNN-3	Z Feature Map	Circular
QNN-4	Z Feature Map	SCA
QNN-5	Z Feature Map	Reverse Linear
QNN-6	Z Feature Map	Pairwise

To further analyze computational complexity, we report the number of single-qubit gates, two-qubit gates, and total gates for each configuration in Table 3. Since two-qubit gates are more computationally expensive, they are expected to significantly influence simulation time.

Table 3 Quantum gate counts for each QNN configuration.

QNN Config	Feature Map	Ansatz	Single-Qubit Gates (Feature + Ansatz)	Two-Qubit Gates (Feature + Ansatz)	Total Gates
QNN-1	Z	Full	16 + 12 = 28	0 + 12 = 12	40
QNN-2	Z	Linear	16 + 12 = 28	0 + 6 = 6	34
QNN-3	Z	Circular	16 + 12 = 28	0 + 8 = 8	36
QNN-4	Z	SCA	16 + 12 = 28	0 + 8 = 8	36
QNN-5	Z	Reverse Linear	16 + 12 = 28	0 + 6 = 6	34
QNN-6	Z	Pairwise	16 + 12 = 28	0 + 6 = 6	34

3.6 Measurement and Optimization

After quantum processing, measurement operations convert quantum states into classical outputs, which are used to compute the loss function. Model parameters are optimized using classical optimizers. In this study, we employed standard optimization techniques (e.g., ADAM and L-BFGS-B) to minimize prediction error. Detailed mathematical formulations are provided in Appendix A.

3.7 Classical Models

For benchmarking, we compared QNN performance against three classical regression models: k-Nearest Neighbors (kNN), Decision Tree Regressor (DTR), and Linear Regression (LR). These models were selected based on their prior effectiveness on the same dataset [22]. Default parameters were used for LR and DTR, while kNN was configured with $k = 5$ using the Minkowski distance metric. All models were evaluated on identical training and test splits across varying dataset sizes.

4 Experimental Results and Discussion

This section presents and analyzes the empirical results of our experiments in a structured and comparative manner. We begin by evaluating six distinct Quantum Neural Network (QNN) configurations, focusing on their predictive performance, stability, and scalability across varying dataset sizes. Subsequently, we analyze simulation time performance. We then evaluate three classical Machine Learning (ML) models and conduct a comparative analysis between QNN and classical approaches.

The primary evaluation metrics used throughout this study are the coefficient of determination (R^2) and root mean square error ($RMSE$). In addition to quantitative evaluation, we provide visual comparisons of actual versus predicted values, along with error distribution analyses, to further interpret model performance.

All experiments were conducted on an HP Z1 All-in-One G2 Workstation with 16 GB RAM and an Intel® Xeon® Processor E3-1246 v3, using Qiskit Machine Learning [29] within a Python programming environment. For all QNN configurations, we used L-BFGS-B with 25 iterations and $\epsilon = 1e{-}8$ (the default value).

4.1 Quantum Neural Network Machine Learning Performance

To evaluate the predictive capability of the QNN configurations, we conducted 5-fold cross-validation experiments across four dataset sizes (800, 1600, 2400, and 3200 training samples). Model performance was assessed using R^2 and $RMSE$, summarized as mean and standard deviation values in Tables 4 and 5.

4.1.1 Cross-Validation Performance Summary

Table 4 Summary of 5-fold cross-validation R^2 scores (mean ± std) for QNN configurations across dataset sizes.

Configuration	800	1600	2400	3200
QNN-1	0.91±0.037	0.93±0.008	0.92±0.020	0.92±0.012
QNN-2	0.91±0.031	0.92±0.023	0.90±0.019	0.89±0.032
QNN-3	0.89±0.038	0.91±0.025	0.92±0.024	0.91±0.028
QNN-4	0.89±0.050	0.92±0.014	0.92±0.018	0.92±0.010
QNN-5	0.90±0.028	0.93±0.009	0.92±0.022	0.91±0.017
QNN-6	0.91±0.031	0.92±0.005	0.91±0.015	0.91±0.013

4.1.2 Performance Trends Across Dataset Sizes

A clear trend emerges in the evolution of R^2 scores as the dataset size increases. Across all configurations, performance improves when moving from 800 to 1600 samples, where most QNNs achieve their peak or near-peak accuracy. Beyond this point, further increases in dataset size do not yield consistent improvements. Instead, R^2 values either plateau or slightly decline, indicating diminishing returns. This behavior

Table 5 Summary of 5-fold cross-validation $RMSE$ (kW) (mean ± std) for QNN configurations across dataset sizes.

Configuration	800	1600	2400	3200
QNN-1	210.24±41.89	191.29±11.53	201.04±26.40	205.37±12.25
QNN-2	215.30±34.52	208.35±29.50	223.90±21.38	233.57±31.74
QNN-3	232.16±42.50	215.13±33.38	201.56±31.69	207.64±27.71
QNN-4	225.75±60.70	200.22±16.28	203.24±23.11	206.93±11.00
QNN-5	218.77±30.01	188.18±12.49	201.87±26.30	209.23±17.29
QNN-6	212.32±35.42	199.98±8.14	213.12±16.43	215.47±11.66

suggests that an optimal data-to-model complexity balance exists for the considered QNN architectures.

The $RMSE$ trends complement this observation. As the dataset size increases from 800 to 1600 samples, a noticeable reduction in prediction error is observed across all configurations. However, when the dataset size exceeds 1600 samples, $RMSE$ begins to increase slightly or stabilize, reflecting reduced marginal gains in predictive accuracy. This pattern indicates that while additional data initially enhances model generalization, excessive data relative to model capacity may lead to suboptimal learning efficiency.

It is important to highlight the point of diminishing returns observed at the 1600-sample mark. This transition point occurred when the training data reached 1600 samples, notably half the maximum dataset size (3200 samples) used in these experiments. Future work could involve conducting similar evaluations using a broader range of dataset types and sizes to determine whether this represents a generalizable characteristic of these QNN architectures.

4.1.3 Stability and Robustness Analysis

Model stability, assessed via standard deviations of R^2 and $RMSE$, varies across configurations and dataset sizes. In general, increasing dataset size improves stability up to 1600 samples, as evidenced by reduced variance in both metrics. Beyond this point, stability trends become less consistent.

Among all configurations, QNN-6 demonstrates the most consistent stability, maintaining low standard deviations across all dataset sizes. QNN-5 and QNN-1 also exhibit relatively stable behavior, whereas QNN-2 shows higher variability, particularly at larger dataset sizes. These findings indicate that stability is configuration-dependent and does not necessarily correlate directly with predictive performance.

4.1.4 Overall Predictive Performance

Figure 3 depicts the overall predictive performance of the six QNN configurations using R^2 and $RMSE$ across all dataset sizes. Throughout the tests, QNN-1 was consistently among the top-performing configurations in terms of both metrics. Improvements in both R^2 and $RMSE$ were generally observed as the dataset size increased to 1600

samples. Beyond this point, further increases did not typically yield additional performance gains and, in some cases, led to a decline. For instance, the R^2 scores for QNN-2 and QNN-5 tended to decrease with larger datasets after initially peaking at 1600 samples. Concerning $RMSE$, a general trend of increasing error was observed for all configurations when the dataset size exceeded 1600 samples.

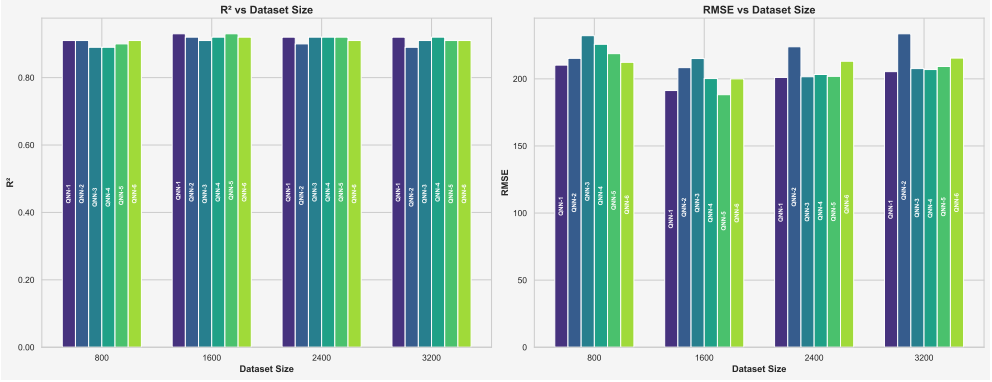


Fig. 3 Average R^2 and $RMSE$ from cross-validation for each QNN configuration across different dataset sizes.

4.1.5 Loss Function Analysis

Figure 4 presents the average training loss per iteration for each QNN configuration across the four dataset sizes. Across all configurations, the loss function decreases significantly within the first 5–10 iterations, indicating effective early-stage learning. After this initial drop, loss values stabilize with only minor fluctuations, demonstrating that QNNs do not require more than approximately 15 iterations to converge for this prediction problem. At the smallest dataset size (800), higher variance and occasional sharp spikes were observed (e.g., QNN-3 and QNN-5), which may indicate overfitting or training instability. For medium datasets (1600 and 2400), convergence was smoother, and for the largest dataset (3200), all configurations achieved the best stability and lowest final loss values.

To quantitatively assess learning stability, we developed an empirical stability score (SC) computed from three metrics averaged over the four dataset sizes: the standard deviation of loss (SD), the maximum spike value after the 10^{th} iteration (MS), and the final loss value at the 25^{th} iteration (FL). Each metric was normalized to $[0, 1]$ using min-max normalization, and the stability score was calculated as:

$$\text{SC} = \text{normalized(SD)} + \text{normalized(MS)} + \text{normalized(FL)} \tag{2}$$

Table 6 summarizes the stability metrics for each QNN configuration. QNN-6 was the most stable model across all metrics, followed by QNN-5 and QNN-1. QNN-2 exhibited the worst stability, with the highest SC score.

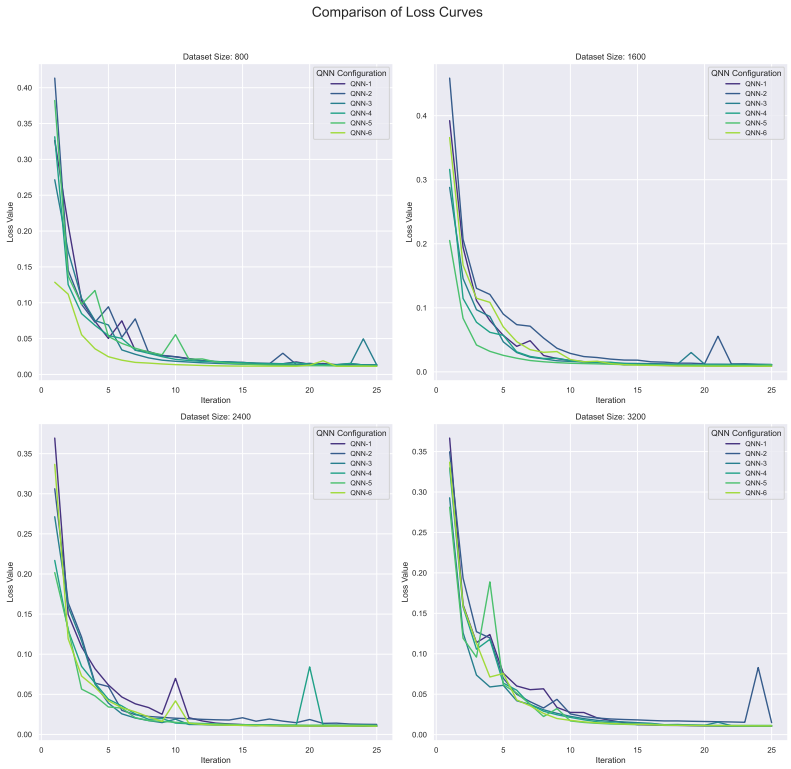


Fig. 4 Average training loss per iteration for each QNN configuration across different dataset sizes, based on 5-fold cross-validation.

Table 6 QNN stability ranking based on average standard deviation (SD), maximum spike (MS), and final loss (FL) values. Lower SC indicates higher stability.

QNN Configuration	SD	MS	FL	SC	Stability Rank
QNN-6	0.002	0.003	0.010	**0.07**	1
QNN-5	0.002	0.002	0.011	0.38	2
QNN-1	0.003	0.004	0.011	0.41	3
QNN-3	0.004	0.015	0.011	1.26	4
QNN-4	0.006	0.020	0.011	1.61	5
QNN-2	0.009	0.033	0.013	3.00	6

These results confirm that even with limited data, QNNs can learn effectively, albeit with some instability. As the dataset size increases, models achieve greater stability, revealing a trade-off between model stability and data requirements. Owing to the long training times of QNN configurations, hyperparameter optimization was not investigated in this study and is left for future work.

4.2 Quantum Neural Network Simulation Time Performance

Current QNN implementations rely on classical optimizers; therefore, the limitations of classical hardware directly affect training times. In this section, we investigate training time from two perspectives: the effect of dataset size and the effect of quantum circuit complexity.

4.2.1 Training Times and Scalability

Table 7 presents the average training times for each QNN configuration across the four dataset sizes. QNN-1 consistently requires the longest training time regardless of dataset size, while QNN-5 and QNN-6 are the most time-efficient configurations.

Table 7 Average training times (minutes) for QNN configurations over 5-fold cross-validation across different dataset sizes.

	Dataset size			
QNN Configuration	**800**	**1600**	**2400**	**3200**
QNN-1	26.33	51.92	77.52	104.68
QNN-2	23.89	47.44	72.03	94.80
QNN-3	23.71	47.43	72.18	96.83
QNN-4	24.65	47.57	73.34	96.17
QNN-5	23.10	47.54	68.62	92.53
QNN-6	23.51	45.62	68.65	92.71

Figure 5 provides a visual representation of these training times as a function of dataset size. To quantify the scalability trend, we fit a linear regression model of the form:

$$\text{Time} = a \times \text{Dataset Size} + b \tag{3}$$

where the slope a indicates the time increase per additional 800 samples and the intercept b represents initial overheads such as circuit preparation and compilation. The results, summarized in Table 8, confirm that QNN-5 is the most time-efficient configuration and QNN-1 the least efficient. This empirical analysis demonstrates that training time scales linearly with dataset size, indicating a time complexity of $\mathcal{O}(n)$.

4.2.2 Impact of Circuit Complexity

Figure 6 shows the relationship between total gate count, dataset size, and average training time. QNN-1, which has the highest total gate count (40), exhibits the worst time performance, while QNN-2, QNN-5, and QNN-6, which share the lowest gate count (34), exhibit the best time performance.

Notably, QNN configurations with the same total gate count exhibit similar time performance regardless of differences in their circuit geometries or entanglement strategies. A linear increase in runtime was observed as the gate count increased. Consequently, the total gate count is a critical parameter that must be considered when designing quantum circuits for QML simulations.

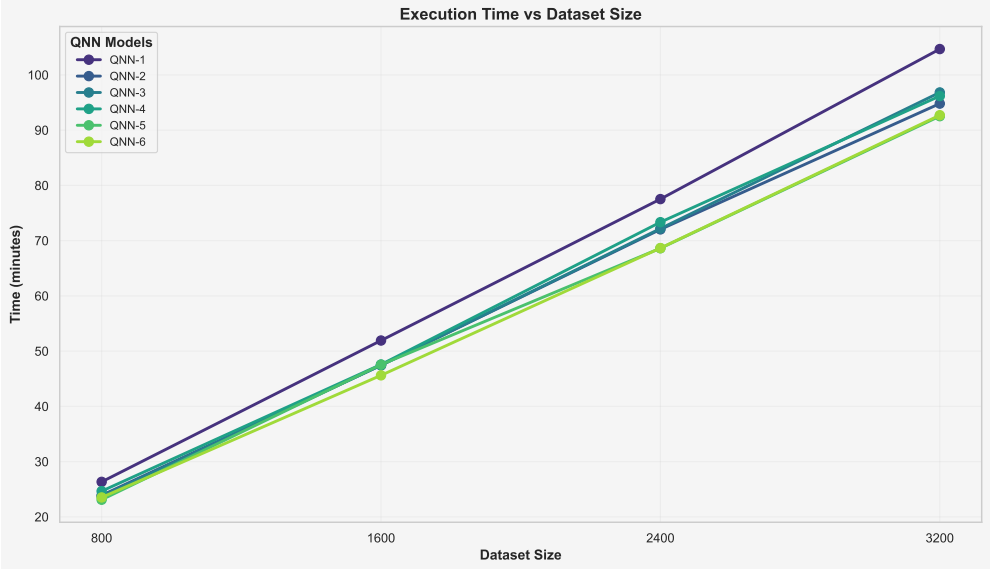


Fig. 5 Training time of each QNN configuration as a function of dataset size.

Table 8 Time efficiency ranking of QNN configurations based on a linear regression of training time versus dataset size.

QNN	Slope	Intercept	Rank
QNN-5	2.87e−02	0.60	1
QNN-6	2.88e−02	−0.03	2
QNN-2	2.97e−02	0.21	3
QNN-4	3.00e−02	0.35	4
QNN-3	3.05e−02	−0.99	5
QNN-1	3.26e−02	−0.05	6

4.3 Classical Model Performance

To benchmark the QNN configurations, classical machine learning models (kNN, DTR, and LR) were evaluated using the same cross-validation framework. Table 9 summarizes their performance.

Across all dataset sizes, classical models show improved performance with increasing data. DTR performs best at medium dataset sizes, while kNN demonstrates superior scalability, achieving the strongest performance at larger dataset sizes. LR consistently underperforms relative to kNN and DTR, particularly in terms of $RMSE$, although it maintains lower variance. Figure 7 provides a visual summary of these trends.

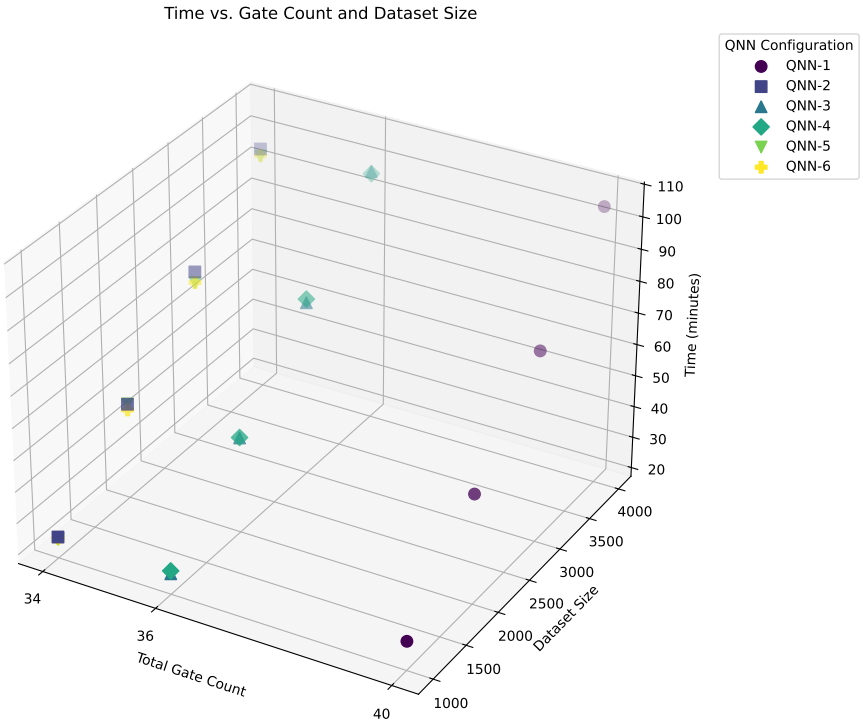


Fig. 6 Relationship between total quantum gate count, dataset size, and average training time for each QNN configuration.

4.4 Comparative Analysis

To directly compare QNN configurations with classical ML models, we used hold-out test datasets of sizes 200, 400, 600, and 800. Table 10 provides the quantitative results.

For all dataset sizes, the quantum approaches consistently achieved the lowest $RMSE$ values and R^2 scores that were equal to or better than those of classical approaches. Among the quantum models, QNN-3 scaled best with increasing dataset size, achieving the highest R^2 (0.94) and lowest $RMSE$ (174.67 kW) on the largest test set. Compared to the best classical model (kNN, with $RMSE$ of 182.99 kW), QNN-3 provides a 4.5% reduction in prediction error. Classical approaches initially lag behind QNN models at smaller dataset sizes (200 samples) but improve as dataset size increases; only at 600 samples do some classical models begin to match several QNN configurations.

Scatter plots comparing actual versus predicted values are provided in Appendix C (Figures C14–C17), and temporal consistency plots are provided in Appendix D (Figures D18–D21). The temporal consistency analysis revealed that while QNN models generally produce fewer errors, their erroneous predictions occasionally include

Table 9 Summary of 5-fold cross-validation performance of classical models (mean ± std).

Dataset size	Model	R^2	RMSE (kW)
800	kNN	0.80±0.05	313.44±35.33
	DTR	0.85±0.07	267.37±73.29
	LR	0.86±0.03	263.30±32.93
1600	kNN	0.90±0.01	233.44±8.10
	DTR	0.93±0.03	193.78±35.94
	LR	0.90±0.01	231.82±11.66
2400	kNN	0.90±0.02	226.78±18.19
	DTR	0.91±0.03	208.66±42.44
	LR	0.88±0.03	247.85±29.25
3200	kNN	0.92±0.013	204.18±12.68
	DTR	0.90±0.017	220.86±16.10
	LR	0.88±0.014	247.36±11.00

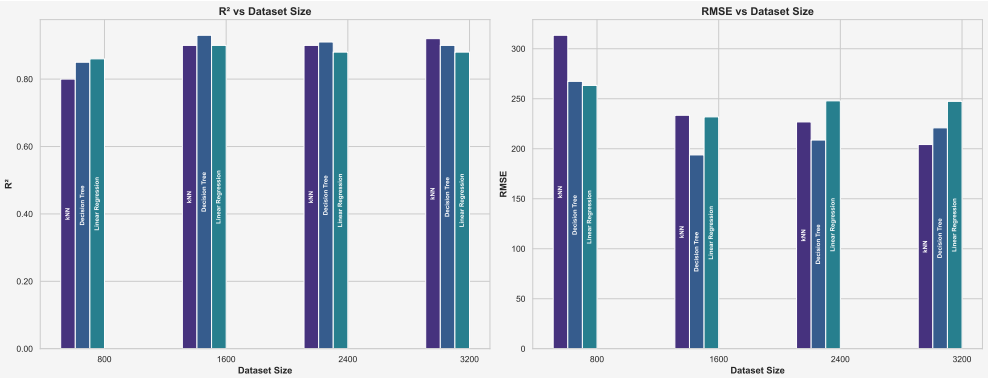


Fig. 7 Average R^2 and $RMSE$ from cross-validation for classical ML models across different dataset sizes.

negative values, which are not physically possible for wind turbine power output. LR exhibits similar behavior, whereas kNN and DTR do not. Future research should investigate methods for enforcing physical constraints, such as applying ReLU activation functions or designing loss functions that penalize negative outputs.

4.4.1 Error Distribution Analysis

Figures 8–11 present comparative error distributions for all models. The quantum approaches exhibited more symmetric and concentrated error distributions than classical models. For the smallest dataset (200 samples), QNN-1 and QNN-5 achieved the tightest error spreads with low bias. As the dataset size increased, QNN-3 emerged as the most stable model, and at the largest dataset (800 samples), QNN-1 and QNN-3 demonstrated superior predictive stability. Among classical models, kNN exhibited

Table 10 Performance comparison of QNN and classical models on the hold-out test set across different dataset sizes.

Approach		Dataset size							
		200		400		600		800	
		R^2	$RMSE$	R^2	$RMSE$	R^2	$RMSE$	R^2	$RMSE$
Quantum	QNN-1	**0.92**	**192.16**	0.89	219.87	0.93	185.11	0.93	184.65
	QNN-2	0.91	209.62	0.88	230.38	0.89	240.41	0.93	186.38
	QNN-3	0.90	216.71	**0.90**	**211.62**	**0.94**	**177.81**	**0.94**	**174.67**
	QNN-4	0.91	204.94	0.90	213.11	0.92	197.49	0.94	179.86
	QNN-5	0.92	201.53	0.89	219.36	0.93	194.51	0.93	186.16
	QNN-6	0.92	201.74	0.89	227.90	0.93	194.99	0.92	197.26
Classical	kNN	0.83	288.96	0.88	233.84	0.93	188.32	0.94	182.99
	DTR	0.88	239.00	0.86	251.55	0.90	222.13	0.92	197.32
	LR	0.89	235.30	0.85	258.54	0.90	229.47	0.90	232.49

nearly zero bias. Table 11 summarizes the top-performing models based on error bias and variance.

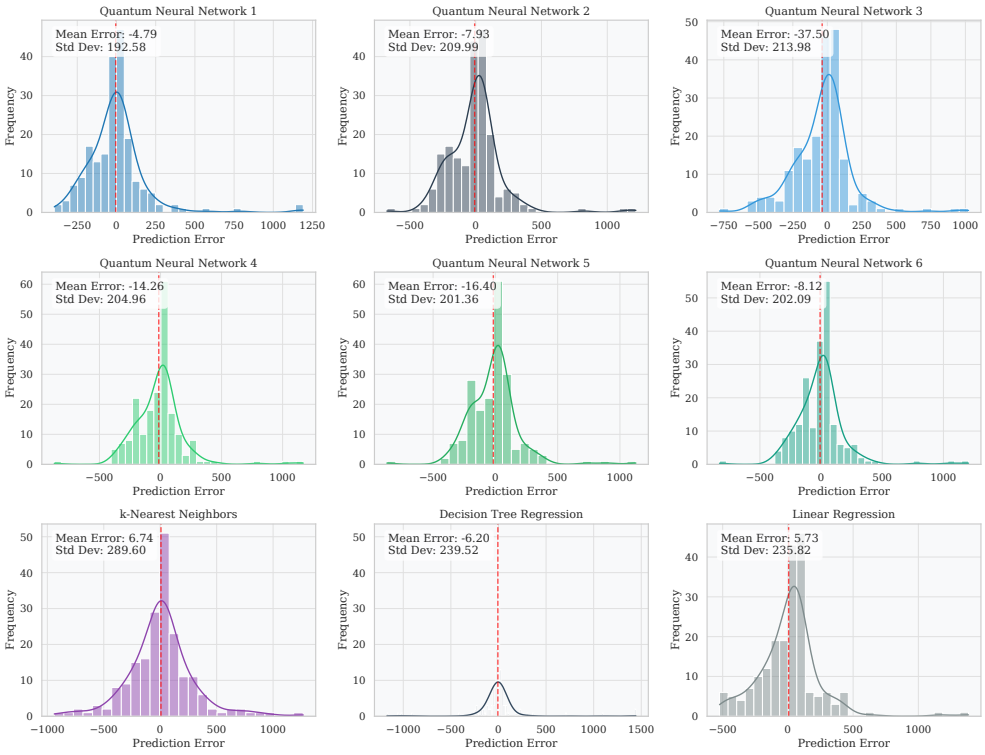


Fig. 8 Error distribution comparing QNN and classical models for a test set of 200 samples.

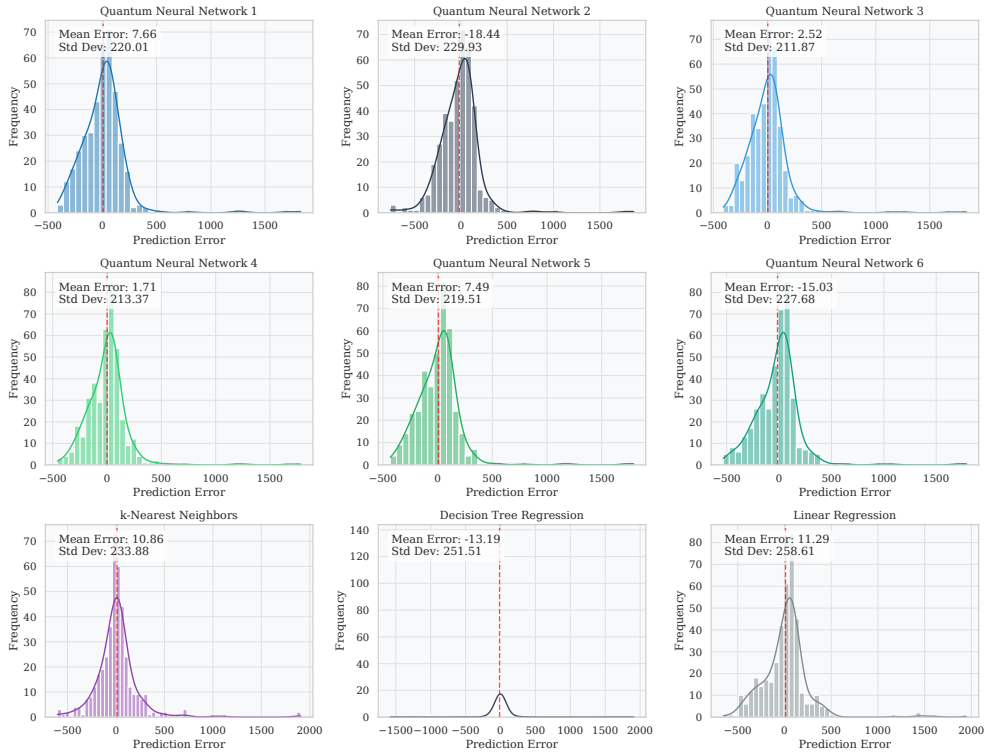


Fig. 9 Error distribution comparing QNN and classical models for a test set of 400 samples.

Table 11 Model ranking by prediction error distribution. The evaluation considers both mean error (bias) and standard deviation (variance).

Dataset Size	1st Rank	2nd Rank	3rd Rank
200	QNN-5	QNN-1	QNN-6
400	QNN-4	QNN-3	QNN-5
600	QNN-3	QNN-5	QNN-1
800	QNN-3	QNN-1	kNN

4.5 Summary of Experimental Findings

Our experiments provide a comprehensive assessment of six QNN configurations across four dataset sizes using 5-fold cross-validation. The key findings are as follows. QNN configurations achieved prediction accuracy with R^2 values of up to 0.94, with peak cross-validation performance at 1600 samples; further data increases did not improve results. This suggests an optimal data-to-model complexity ratio may exist for these ansatz configurations. The loss function analysis confirmed rapid convergence within approximately 10 iterations, with larger datasets contributing to more stable training

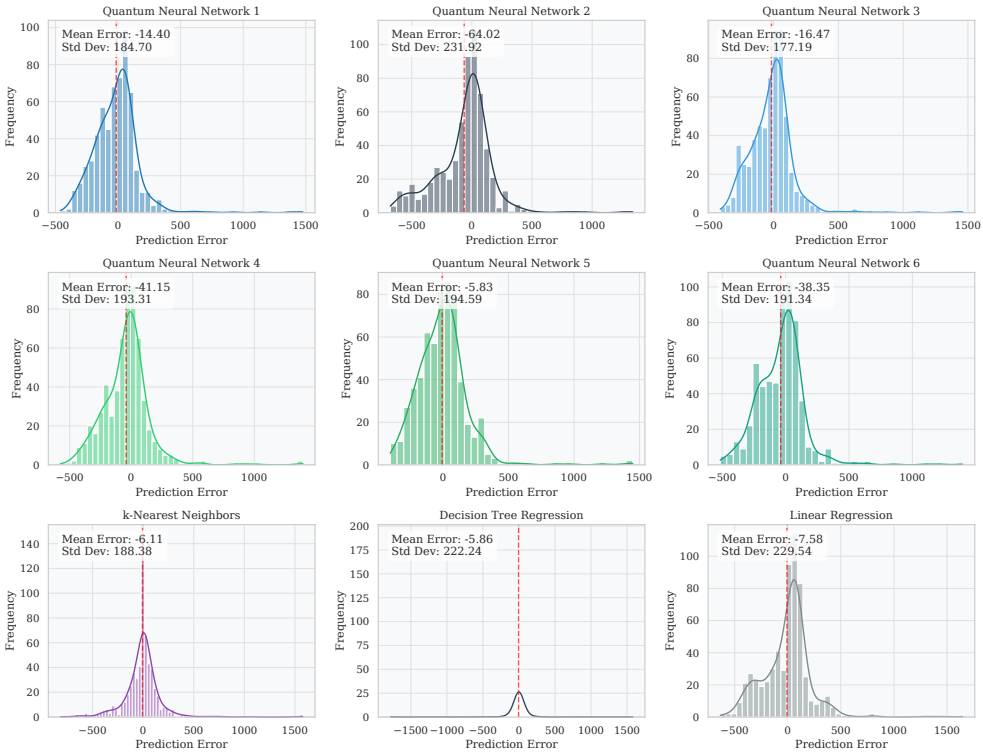


Fig. 10 Error distribution comparing QNN and classical models for a test set of 600 samples.

dynamics. QNN-6 was identified as the most stable configuration overall. Simulation time scaled linearly with dataset size ($\mathcal{O}(n)$), with the total gate count as the primary driver; QNN-5 and QNN-6 were the most time-efficient models.

In the comparative analysis on hold-out test sets, quantum models demonstrated superior performance for smaller datasets. QNN-3 yielded an R^2 of 0.94 and $RMSE$ of 174.67 kW on the largest test set, compared to the best classical model (kNN) at R^2 of 0.94 and $RMSE$ of 182.99 kW. These findings quantitatively demonstrate the potential of QNNs as a competitive alternative to classical approaches for real-world prediction problems.

5 Conclusion and Future Work

In this study, we conducted a systematic benchmark of six Quantum Neural Network (QNN) configurations against three classical machine learning approaches: k-nearest neighbors (kNN), decision tree regression (DTR), and linear regression (LR), using a real-world wind turbine power forecasting dataset. The QNN models were constructed

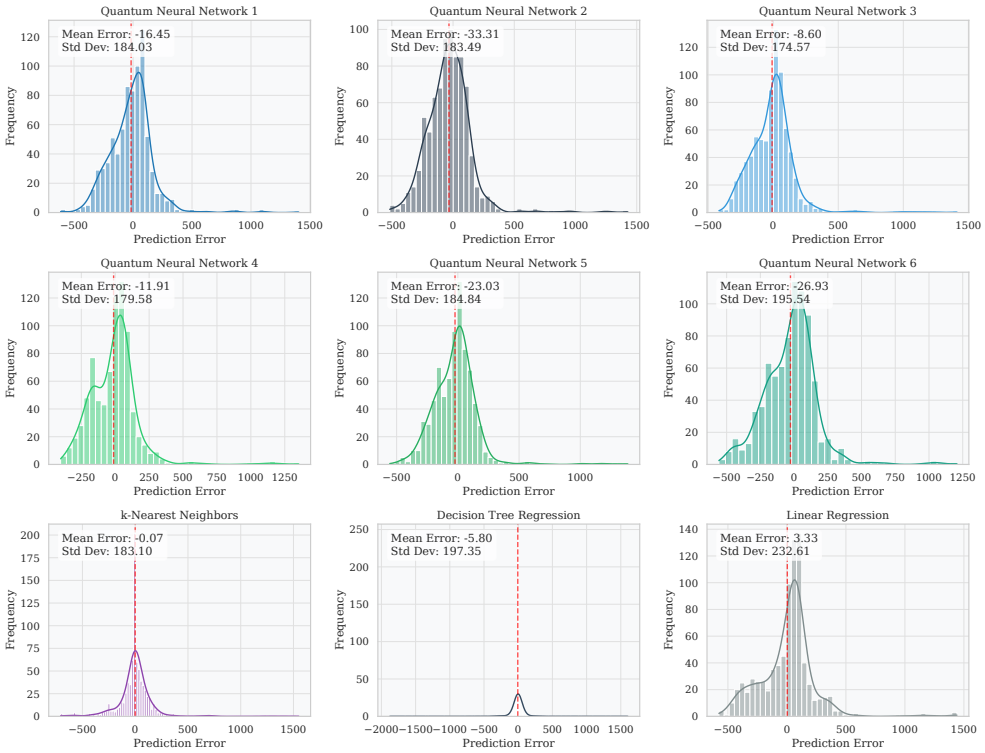


Fig. 11 Error distribution comparing QNN and classical models for a test set of 800 samples.

using a Z feature map combined with diverse ansatz structures, and their performance was evaluated with respect to predictive accuracy, stability, and simulation time efficiency across varying dataset sizes.

The experimental results demonstrate that QNN configurations are competitive with classical approaches, particularly in data-limited scenarios. Among the evaluated models, QNN-3 achieved the strongest overall performance, reaching an R^2 score of 0.94 and an $RMSE$ of 174.67 kW on the largest test set, which is comparable to the best-performing classical model (kNN), which achieved an R^2 of 0.94 and an $RMSE$ of 182.99 kW. These results indicate that QNNs can provide similar predictive capability to classical models under the conditions explored in this study.

A key finding of this work is the identification of a performance peak at a dataset size of 1600 samples, beyond which further increases in data did not lead to consistent improvements in either R^2 or $RMSE$. This observation suggests the presence of an optimal data-to-model complexity ratio for the considered QNN architectures, highlighting a potential limitation in their scalability.

In addition, while QNN models achieved competitive accuracy, they occasionally produced physically invalid predictions, such as negative power outputs. This limitation highlights the need for incorporating domain-specific constraints into the modeling

process, either through modified loss functions, output activation mechanisms, or post-processing techniques. Addressing this issue is essential for improving the reliability of QNNs in real-world applications.

From a computational perspective, we demonstrated that the training time of QNN configurations scales linearly with dataset size, exhibiting a time complexity of $\mathcal{O}(n)$. Furthermore, simulation time was strongly influenced by quantum circuit complexity, particularly the total gate count, with simpler circuits (e.g., QNN-5 and QNN-6) achieving better time efficiency. These findings provide practical guidance for designing computationally efficient QNN architectures in the current noisy intermediate-scale quantum (NISQ) era.

We also analyzed the training dynamics and stability of QNN models, observing rapid convergence within approximately 10–15 iterations across all configurations. Larger datasets generally contributed to improved stability and smoother convergence behavior. Among all configurations, QNN-6 demonstrated the most consistent stability across different dataset sizes.

Despite these promising results, several limitations should be acknowledged. First, hyperparameter optimization was not explored for either quantum or classical models due to computational constraints. Second, the classical baselines considered in this study are limited to relatively simple models. More advanced approaches commonly used in regression and time-series forecasting, such as gradient boosting methods (e.g., XGBoost, LightGBM), support vector regression (SVR), and deep learning models (e.g., LSTM networks), were not included. Incorporating these stronger baselines is essential for a more comprehensive evaluation of QNN performance and will be a key focus of future work.

In future research, we aim to extend this study by incorporating advanced classical baselines, exploring hyperparameter optimization strategies, and investigating techniques for enforcing physical constraints in QNN predictions. Additionally, evaluating these models on larger and more diverse datasets, as well as leveraging emerging hardware-accelerated quantum simulators and real quantum devices, will be important steps toward assessing the practical applicability of quantum machine learning in real-world scenarios.

Overall, this study provides a structured and empirical evaluation of QNN-based approaches for regression tasks and highlights their potential as a complementary alternative to classical machine learning methods, particularly in scenarios with limited data and constrained model complexity.

Acknowledgements. The authors would like to acknowledge that this paper is submitted in partial fulfillment of the requirements for the PhD degree at Yildiz Technical University.

Declarations

This research received no external funding. All authors certify that they have no affiliations with or involvement in any organization or entity with any financial or non-financial interest in the subject matter or materials discussed in this manuscript.

Funding. Open access funding was provided by the Scientific and Technological Research Council of Türkiye (TÜBİTAK).

Conflict of Interest. The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Ethics approval and consent to participate. Not applicable.

Data Availability. Data will be made available upon reasonable request.

Appendix A Mathematical Background

This appendix provides the mathematical formulations of the quantum gates, feature maps, and ansatz constructions used in this study.

A.1 Quantum Gates

The Hadamard (H) gate creates superposition and is defined as:

$$H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \tag{A1}$$

The Z-axis rotation gate $R_z(\theta)$ is defined as:

$$R_z(\theta) = \begin{bmatrix} e^{-i\theta/2} & 0 \\ 0 & e^{i\theta/2} \end{bmatrix} \tag{A2}$$

An equivalent phase shift gate $P(\varphi)$ is given by:

$$P(\varphi) = \begin{bmatrix} 1 & 0 \\ 0 & e^{i\varphi} \end{bmatrix} \tag{A3}$$

The Y-axis rotation gate $R_y(\theta)$ is defined as:

$$R_y(\theta) = \begin{bmatrix} \cos(\theta/2) & -\sin(\theta/2) \\ \sin(\theta/2) & \cos(\theta/2) \end{bmatrix} \tag{A4}$$

The controlled-NOT (CNOT) gate acting on two qubits is:

$$\text{CNOT} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \tag{A5}$$

A.2 Z Feature Map

Given a classical feature vector $x = (x_1, x_2, \ldots, x_n)$, the Z Feature Map encodes the data into a quantum state using the following steps:

1. Apply Hadamard gates to all qubits: $H^{\otimes n}$
2. Apply Z-axis rotations: $R_z(2x_j)$ for each qubit j

The resulting quantum state is:

$$|\psi(x)\rangle = \bigotimes_{j=1}^{n} \frac{1}{\sqrt{2}} \left(|0\rangle + e^{i2x_j} |1\rangle\right) \tag{A6}$$

A.3 Ansatz Construction

The variational quantum circuit (ansatz) prepares a parameterized quantum state using a unitary operator $U(\boldsymbol{\theta})$ acting on an initial state $|\psi_0\rangle$:

$$|\psi(\boldsymbol{\theta})\rangle = U(\boldsymbol{\theta}) |\psi_0\rangle \tag{A7}$$

The ansatz is constructed using alternating layers of single-qubit rotations and entangling gates:

$$U(\boldsymbol{\theta}) = \prod_{l=1}^{L} \left(\bigotimes_{j=1}^{n} R_{\phi_j}^{(j)}(\theta_{l,j}) \cdot E_l \right) \tag{A8}$$

where:

- $R_{\phi_j}^{(j)}(\theta_{l,j})$ denotes a rotation gate applied to qubit j,
- $\phi_j \in \{x, y, z\}$ specifies the rotation axis,
- E_l represents the entangling operation in layer l.

A.4 Cost Function

The objective of training is to find parameters $\boldsymbol{\theta}$ that minimize the expectation value of a cost Hamiltonian H:

$$E(\boldsymbol{\theta}) = \langle \psi(\boldsymbol{\theta}) | H | \psi(\boldsymbol{\theta}) \rangle \tag{A9}$$

This expectation value is evaluated after measurement and used by a classical optimizer to update the circuit parameters iteratively.

Appendix B Quantum Circuits Used in This Study

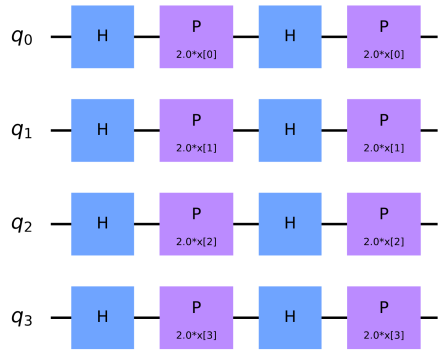


Fig. B1 Quantum circuit for the Z Feature Map with two repetitions.

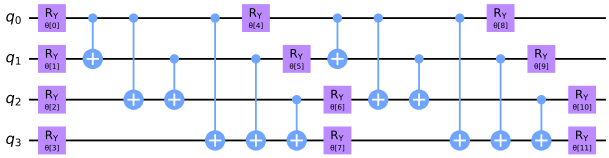


Fig. B2 Ansatz with full entanglement.

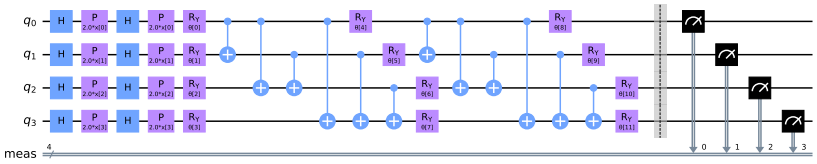


Fig. B3 Complete quantum circuit for QNN-1.

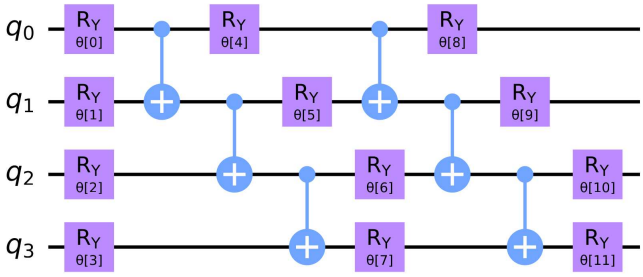


Fig. B4 Ansatz with linear entanglement.

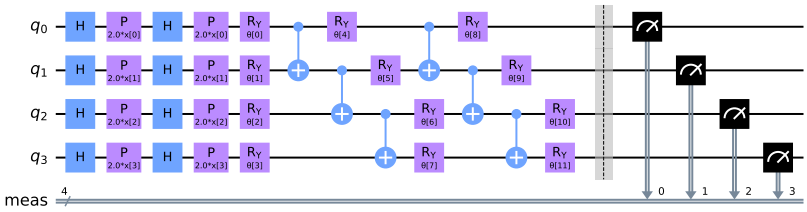


Fig. B5 Complete quantum circuit for QNN-2.

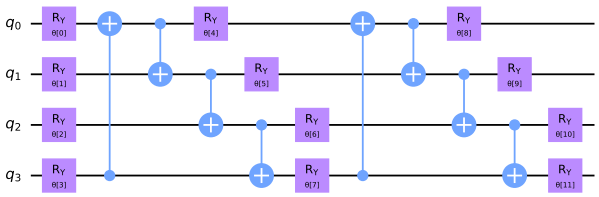


Fig. B6 Ansatz with circular entanglement.

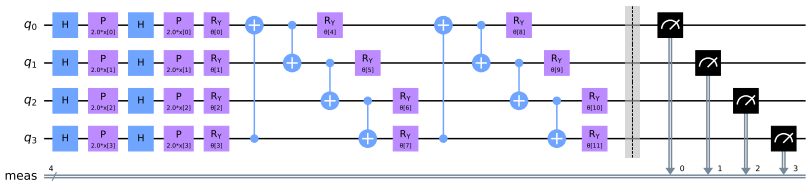


Fig. B7 Complete quantum circuit for QNN-3.

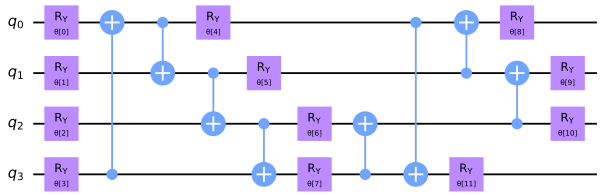


Fig. B8 Ansatz with Shifted-Circular-Alternating (SCA) entanglement.

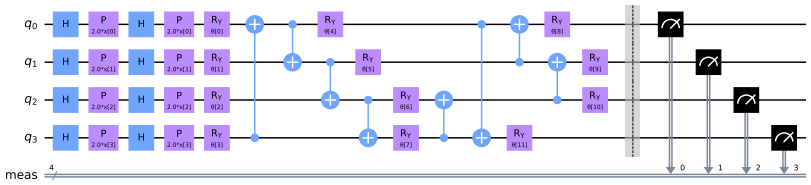


Fig. B9 Complete quantum circuit for QNN-4.

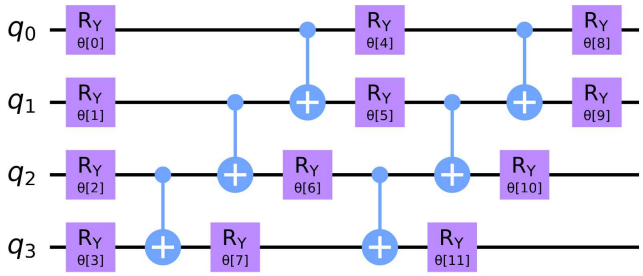


Fig. B10 Ansatz with reverse linear entanglement.

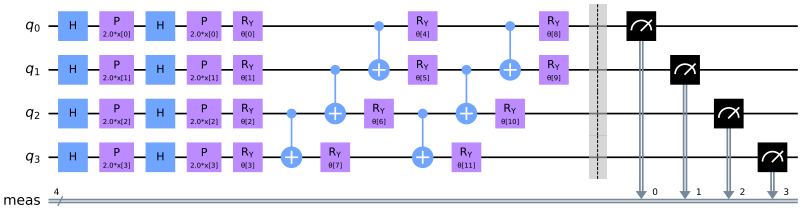


Fig. B11 Complete quantum circuit for QNN-5.

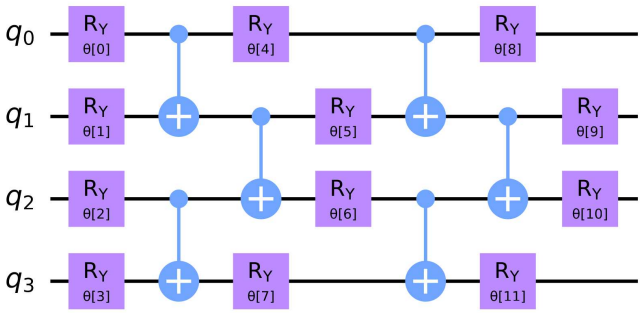


Fig. B12 Ansatz with pairwise entanglement.

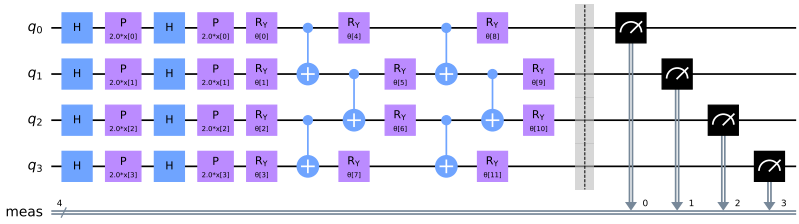


Fig. B13 Complete quantum circuit for QNN-6.

Appendix C Actual vs. Predicted Scatter Plots

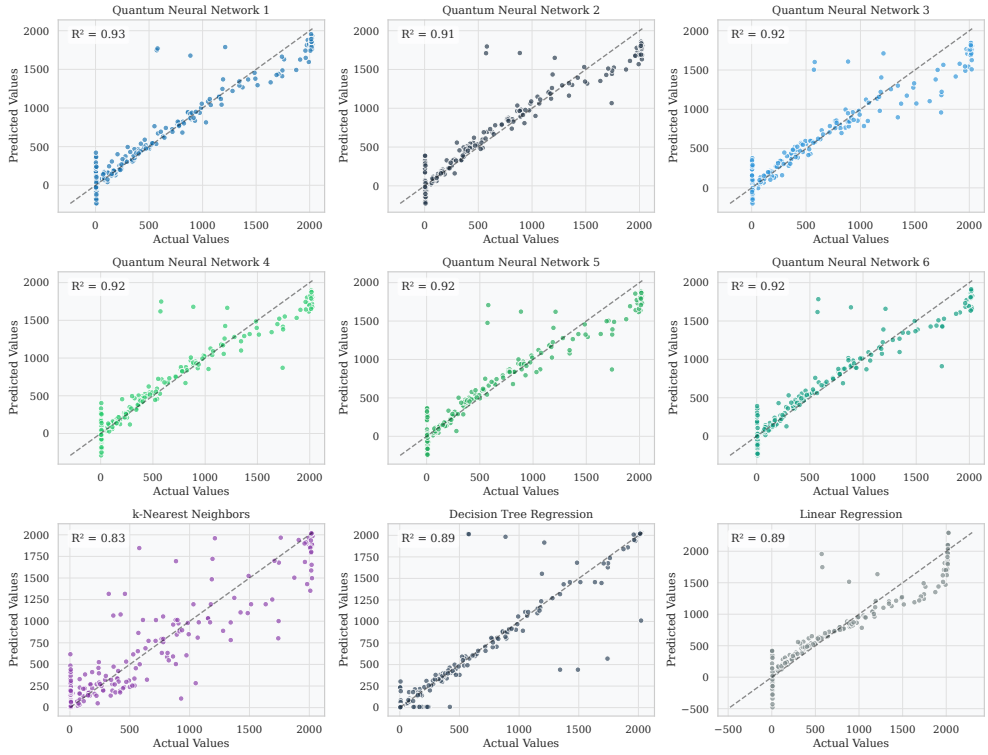


Fig. C14 Scatter plot of actual vs predicted values comparing QNN and classical models for a test set of 200 samples.

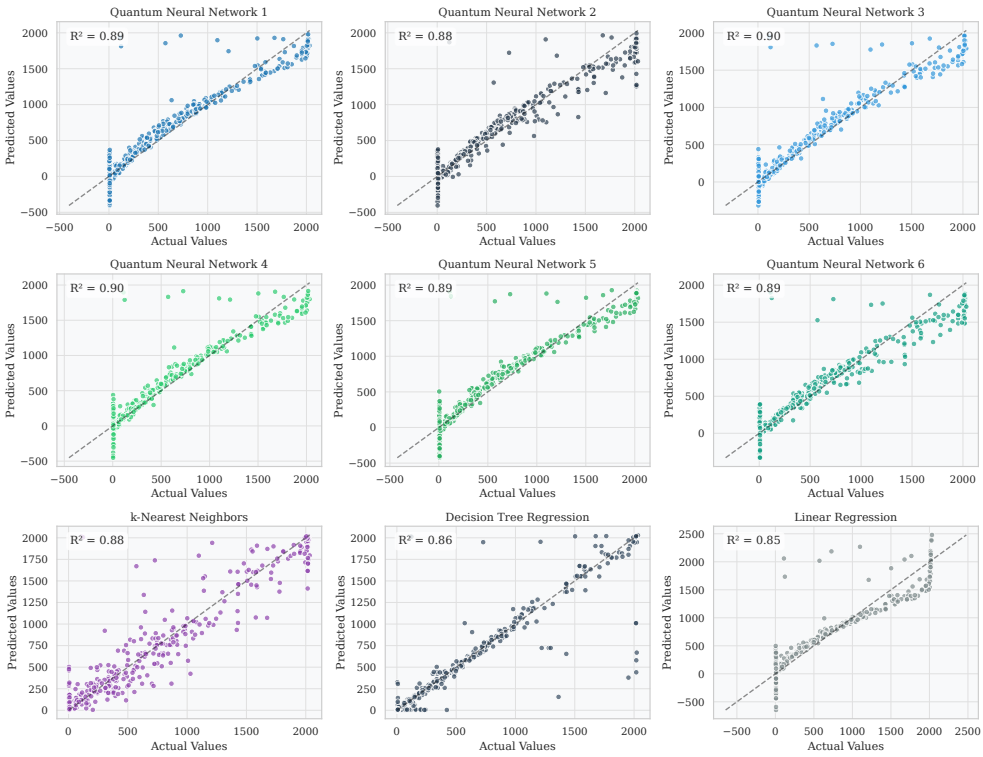


Fig. C15 Scatter plot of actual vs predicted values comparing QNN and classical models for a test set of 400 samples.

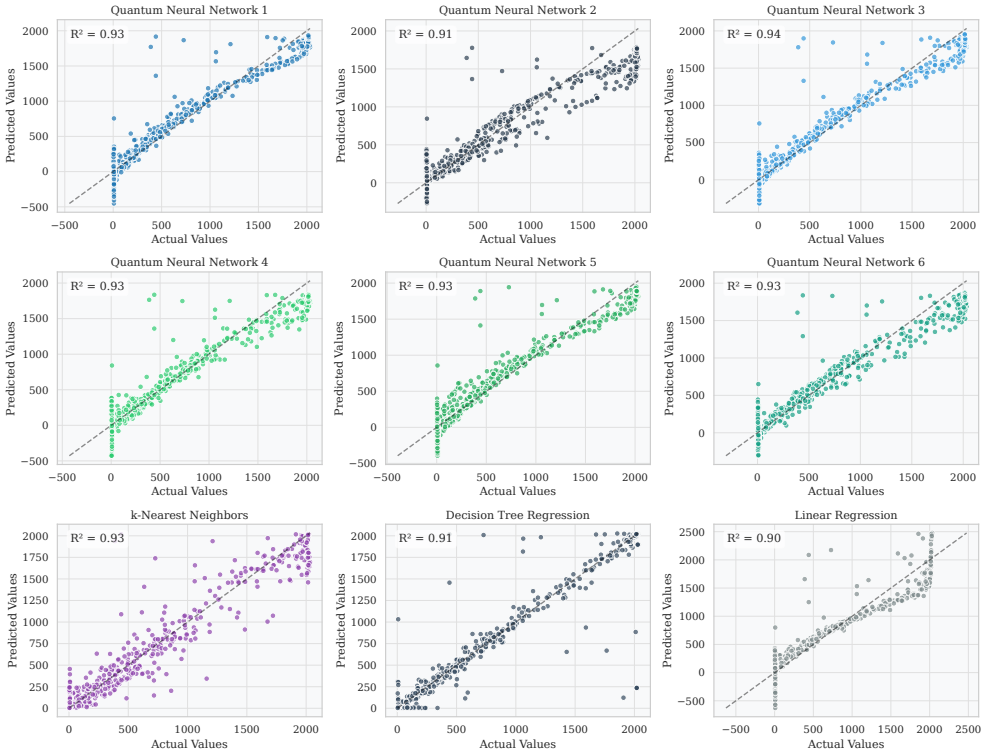


Fig. C16 Scatter plot of actual vs predicted values comparing QNN and classical models for a test set of 600 samples.

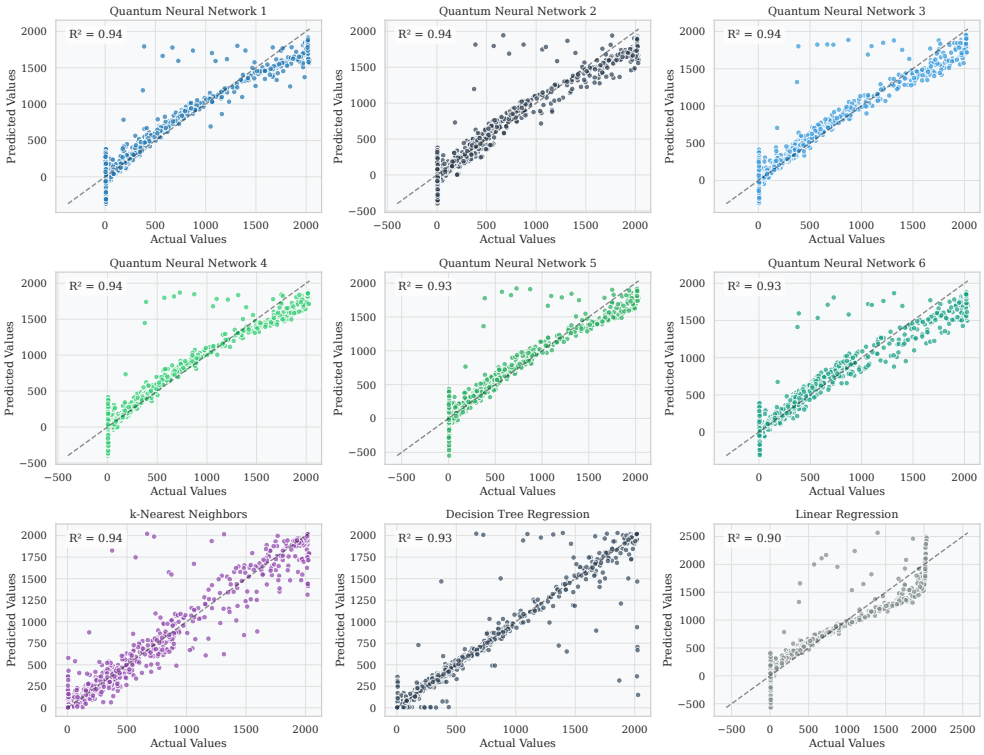


Fig. C17 Scatter plot of actual vs predicted values comparing QNN and classical models for a test set of 800 samples.

Appendix D Temporal Consistency Plots

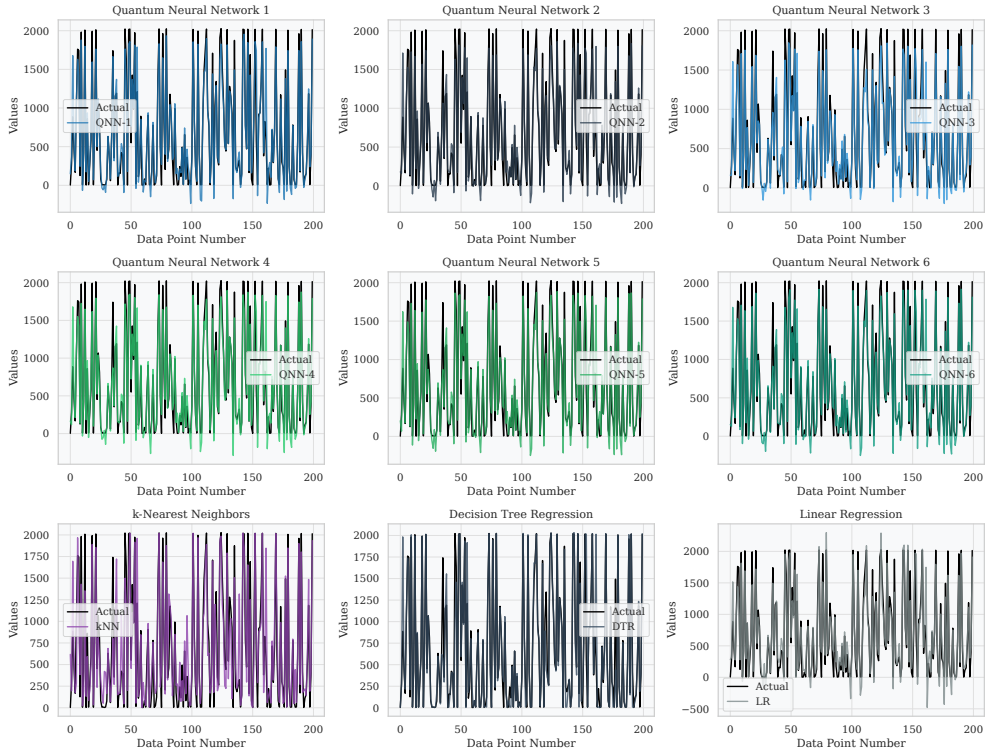


Fig. D18 Line plot of actual vs predicted values comparing QNN and classical models for a test set of 200 samples.

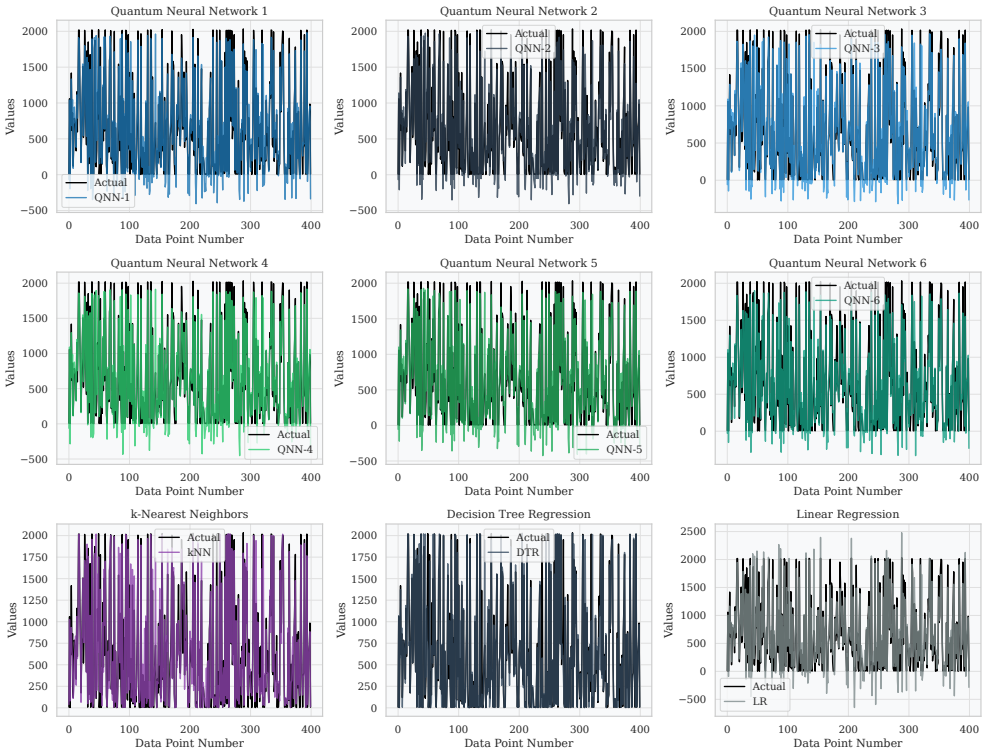


Fig. D19 Line plot of actual vs predicted values comparing QNN and classical models for a test set of 400 samples.

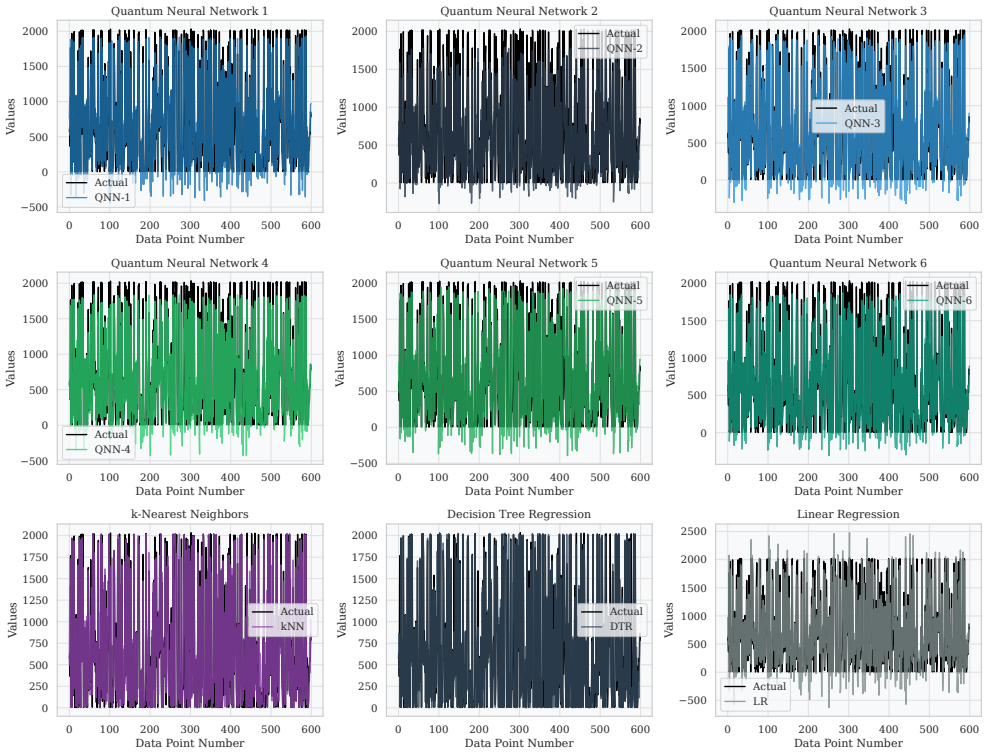


Fig. D20 Line plot of actual vs predicted values comparing QNN and classical models for a test set of 600 samples.

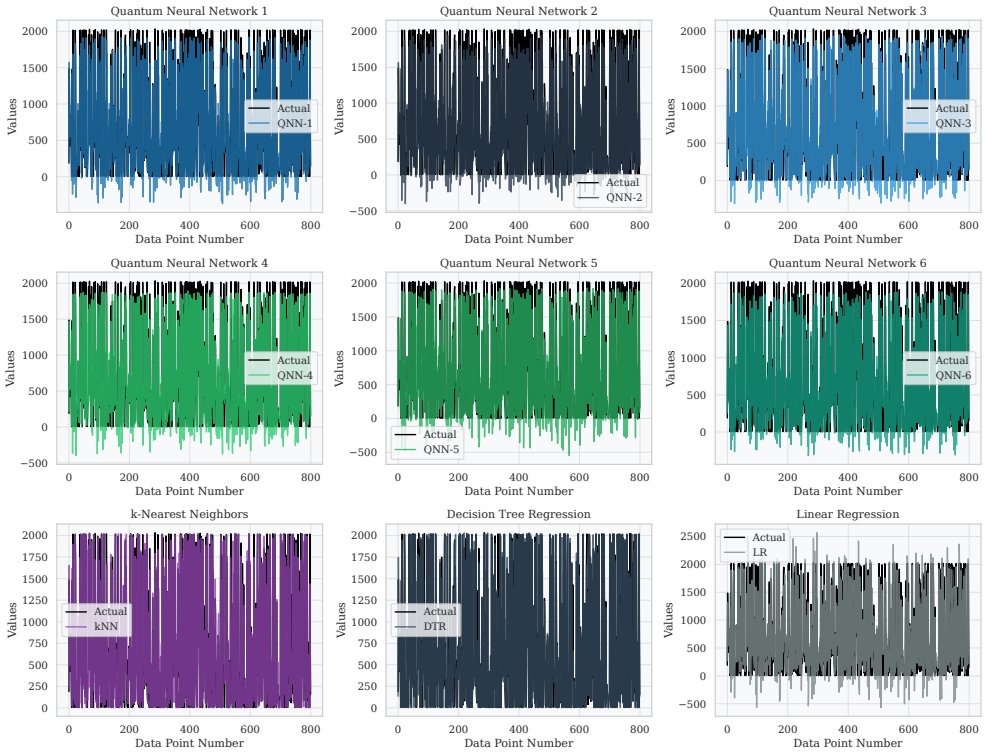


Fig. D21 Line plot of actual vs predicted values comparing QNN and classical models for a test set of 800 samples.

References

[1] Collaborators, G.Q.A..: Quantum error correction below the surface code threshold. NATURE (2024) https://doi.org/10.1038/s41586-024-08449-y

[2] Microsoft Azure Quantum: Interferometric single-shot parity measurement in inas-al hybrid devices. NATURE **638**(8051) (2025) https://doi.org/10.1038/s41586-024-08445-2

[3] Tom, D., Svore, K.: How Microsoft and Quantinuum Achieved Reliable Quantum Computing. Microsoft Azure Quantum Blog (2024). https://azure.microsoft.com/en-us/blog/quantum/2024/04/03/how-microsoft-and-quantinuum-achieved-reliable-quantum-computing/ Accessed 2024-04-03

[4] Kyriienko, O., Magnusson, E.B.: Unsupervised quantum machine learning for fraud detection. arXiv preprint arXiv:2208.01203 (2022)

[5] Sakhnenko, A., O'Meara, C., Ghosh, K.J., Mendl, C.B., Cortiana, G., Bernabé-Moreno, J.: Hybrid classical-quantum autoencoder for anomaly detection. Quantum Machine Intelligence **4**(2), 27 (2022) https://doi.org/10.1007/s42484-022-00075-z

[6] Amin, J., Anjum, M.A., Ibrar, K., Sharif, M., Kadry, S., Crespo, R.G.: Detection of anomaly in surveillance videos using quantum convolutional neural networks. Image and Vision Computing **135**, 104710 (2023) https://doi.org/10.1016/j.imavis.2023.104710

[7] Wang, M., Huang, A., Liu, Y., Yi, X., Wu, J., Wang, S.: A quantum-classical hybrid solution for deep anomaly detection. Entropy **25**(3) (2023) https://doi.org/10.3390/e25030427

[8] Ajagekar, A., You, F.: Quantum computing based hybrid deep learning for fault diagnosis in electrical power systems. Applied Energy **303**, 117628 (2021) https://doi.org/10.1016/j.apenergy.2021.117628

[9] Yu, Y., Hu, G., Liu, C., Xiong, J., Wu, Z.: Prediction of solar irradiance one hour ahead based on quantum long short-term memory network. IEEE Transactions on Quantum Engineering **4**, 1–15 (2023) https://doi.org/10.1109/TQE.2023.3271362

[10] Satpathy, S.K., Vibhu, V., Behera, B.K., Al-Kuwari, S., Mumtaz, S., Farouk, A.: Analysis of quantum machine learning algorithms in noisy channels for classification tasks in the iot extreme environment. IEEE Internet of Things Journal **11**(3), 3840–3852 (2024) https://doi.org/10.1109/JIOT.2023.3300577

[11] Safari, A., Badamchizadeh, M.A.: Neuroquman: quantum neural network-based

consumer reaction time demand response predictive management. Neural Computing and Applications **36**(30), 19121–19138 (2024) https://doi.org/10.1007/s00521-024-10201-6

[12] Hangun, B., Eyecioglu, O., Altun, O.: Quantum computing approach to smart grid stability forecasting. In: 2024 12th International Conference on Smart Grid (icSmartGrid), pp. 840–843 (2024). https://doi.org/10.1109/icSmartGrid61824.2024.10578114

[13] Hangun, B., Akpinar, E., Oduncuoglu, M., Altun, O., Eyecioglu, O.: A hybrid quantum-classical machine learning approach to offshore wind farm power forecasting. In: 2024 13th International Conference on Renewable Energy Research and Applications (ICRERA), pp. 1105–1110 (2024). https://doi.org/10.1109/ICRERA62673.2024.10815339

[14] Ajagekar, A., You, F.: Variational quantum circuit based demand response in buildings leveraging a hybrid quantum-classical strategy. Applied Energy **364**, 123244 (2024) https://doi.org/10.1016/j.apenergy.2024.123244

[15] Ju, Y., Sun, G., Chen, Q., Zhang, M., Zhu, H., Rehman, M.U.: A model combining convolutional neural network and lightgbm algorithm for ultra-short-term wind power forecasting. IEEE Access **7**, 28309–28318 (2019) https://doi.org/10.1109/ACCESS.2019.2901920

[16] Shabbir, N., AhmadiAhangar, R., Kütt, L., Iqbal, M.N., Rosin, A.: Forecasting short term wind energy generation using machine learning. In: 2019 IEEE 60th International Scientific Conference on Power and Electrical Engineering of Riga Technical University (RTUCON), pp. 1–4 (2019). https://doi.org/10.1109/RTUCON48111.2019.8982365

[17] Sulaiman, M.H., Mustaffa, Z.: Enhancing wind power forecasting accuracy with hybrid deep learning and teaching-learning-based optimization. Cleaner Energy Systems **9**, 100139 (2024) https://doi.org/10.1016/j.cles.2024.100139

[18] Olcay, K., Gíray Tunca, S., Aríf Özgür, M.: Forecasting and performance analysis of energy production in solar power plants using long short-term memory (lstm) and random forest models. IEEE Access **12**, 103299–103312 (2024) https://doi.org/10.1109/ACCESS.2024.3432574

[19] Marugán, A.P., Márquez, F.P.G., Perez, J.M.P., Ruiz-Hernández, D.: A survey of artificial neural network in wind energy systems. Applied Energy **228**, 1822–1836 (2018) https://doi.org/10.1016/j.apenergy.2018.07.084

[20] Wang, Z., Yu, D., Wu, Z.: Real-time machine-learning-based optimization using input convex long short-term memory network. Applied Energy **377**, 124472 (2025) https://doi.org/10.1016/j.apenergy.2024.124472

[21] Saxena, V., Kumar, N., Manna, S., Rajput, S.K., Agarwal, K.L., Diwania, S., Gupta, V.: Modelling, solution and application of optimization techniques in hres: From conventional to artificial intelligence. Applied Energy **380**, 125047 (2025) https://doi.org/10.1016/j.apenergy.2024.125047

[22] Eyecioglu, O., Hangun, B., Kayisli, K., Yesilbudak, M.: Performance comparison of different machine learning algorithms on the prediction of wind turbine power generation. In: 2019 8th International Conference on Renewable Energy Research and Applications (ICRERA), pp. 922–926 (2019). https://doi.org/10.1109/ICRERA47325.2019.8996541

[23] Denmark, T.U., Laboratory, R.N.: Database on Wind Characteristics. http://www.winddata.com. [Online]

[24] Ranga, D., Rana, A., Prajapat, S., Kumar, P., Kumar, K., Vasilakos, A.V.: Quantum machine learning: Exploring the role of data encoding techniques, challenges, and future directions. Mathematics **12**(21) (2024) https://doi.org/10.3390/math12213318

[25] Hangun, B., Akpinar, E., Altun, O., Eyecioglu, O.: Comparative Analysis of QNN Architectures for Wind Power Prediction: Feature Maps and Ansatz Configurations (2025). https://arxiv.org/abs/2506.14795

[26] Abbas, A., Sutter, D., Zoufal, C., Lucchi, A., Figalli, A., Woerner, S.: The power of quantum neural networks. NATURE COMPUTATIONAL SCIENCE **1**(6), 403–409 (2021) https://doi.org/10.1038/s43588-021-00084-1

[27] Du, Y., Huang, T., You, S., Hsieh, M.-H., Tao, D.: Quantum circuit architecture search for variational quantum algorithms. NPJ QUANTUM INFORMATION **8**(1) (2022) https://doi.org/10.1038/s41534-022-00570-y

[28] Arthur, D., Date, P.: A Hybrid Quantum-Classical Neural Network Architecture for Binary Classification (2022). https://arxiv.org/abs/2201.01820

[29] Sahin, M.E., Altamura, E., Wallis, O., Wood, S.P., Dekusar, A., Millar, D.A., Imamichi, T., Matsuo, A., Mensa, S.: Qiskit Machine Learning: an open-source library for quantum machine learning tasks at scale on quantum hardware and classical simulators (2025). https://arxiv.org/abs/2505.17756