

IJEAST

INTERNATIONAL JOURNAL
OF ENGINEERING APPLIED SCIENCE
AND TECHNOLOGY

VOLUME : 11 ISSUE : 06 Print / Issue Publication Date: October 2026

ISSN : 2455-2143

Indexed In

Google SCHOLAR OPEN ACCESS doi Crossref

WWW.IJEAST.COM

editor@ijeast.com

International Journal of Engineering Applied Sciences and Technology, 2026
Vol. 11, Issue 06, ISSN No. 2455-2143, Pages 30-36
Published Online October 2026 in IJEAST (http://www.ijeast.com)

A HYBRID QUANTUM-CLASSICAL MACHINE LEARNING FRAMEWORK FOR IMPROVED CLASSIFICATION ACCURACY

Shweta Narayan Bamne, Prof M.S. Shegokar
Department of Computer Science and Engineering
College of Engineering and Technology, Akola, India

Abstract—**Quantum computing has emerged as a promising technology for solving complex computational problems by leveraging the principles of quantum mechanics, including superposition and entanglement. Integrating quantum computing with classical machine learning offers a potential approach to improving learning capabilities and classification performance. This research proposes a Hybrid Quantum-Classical Machine Learning Framework that combines classical data preprocessing techniques with parameterized quantum circuits to perform classification tasks. The framework uses classical machine learning components for feature preprocessing and data encoding, while a variational quantum circuit extracts quantum representations of the input data. A classical optimizer updates the quantum circuit parameters iteratively to minimize a predefined classification loss function. The proposed framework is evaluated using benchmark classification datasets and compared with conventional machine learning models, including Support Vector Machines, Logistic Regression, and classical Neural Networks. Performance is assessed using accuracy, precision, recall, F1-score, computational cost, and training time. Experiments also investigate the effects of circuit depth, feature selection, and simulated quantum noise on classification performance. The framework is implemented using the PennyLane quantum simulation library alongside standard machine learning libraries. The study identifies the conditions under which hybrid quantum-classical learning can provide competitive classification performance and determines the practical limitations of current quantum machine learning approaches. The findings contribute to a better understanding of the effectiveness, scalability, and feasibility of hybrid quantum classical models for real-world classification problems.**

Index Terms—**Quantum Computing, Hybrid Quantum Classical Learning, Quantum Machine Learning, Variational Quantum Circuits, Classification, Qiskit, PennyLane, Quantum Optimization.**

I. INTRODUCTION

Machine learning methods for classification have matured to a point where, on most tabular problems, a well-tuned Support Vector Machine or a small neural network is hard to beat. At the same time, quantum computing has moved from theory to small but working hardware. Today's devices hold tens to a few hundred qubits, have limited coherence time, and suffer from gate errors. Preskill described this period as the Noisy Intermediate-Scale Quantum (NISQ) era [1]. Machines of this size cannot run deep fault-tolerant algorithms, so researchers have looked for shorter circuits that still do something useful.

Quantum machine learning is one of the more active directions within this search [2]. The central idea is that a quantum state of n qubits lives in a Hilbert space of dimension 2^n, so a short circuit can map input data into a very large feature space. Havl´ıˇcek et al. demonstrated quantum feature maps for supervised learning on superconducting hardware [3], and Schuld and Killoran showed that such encodings behave like kernel methods in feature Hilbert space [4]. Parameterized, or variational, quantum circuits take this further. Their gateangles are trained like the weights of a neural network, while a classical computer runs the optimizer [5], [6]. This split of work between a quantum processor and a classical one is what we refer to as hybrid quantum-classical learning.

Several circuit designs for classification have appeared.

Farhi and Neven proposed quantum neural networks for binary labels on near-term processors [7]. Mitarai et al. introduced quantum circuit learning with gradients obtained directly from the circuit [8], and Schuld et al. designed circuit-centric classifiers with few parameters [9]. Benedetti et al. surveyed parameterized circuits as machine learning models [10], and Mari et al. combined pretrained classical networks with variational circuits through transfer learning [11]. Abbas et al. argued that quantum neural networks can have higher effective dimension than comparable classical networks [12].

There are known problems, though. McClean et al. proved that randomly initialized deep circuits tend to have exponentially flat loss landscapes, the so-called barren plateaus [13].

International Journal of Engineering Applied Sciences and Technology, 2026
Vol. 11, Issue 06, ISSN No. 2455-2143, Pages 30-36
Published Online October 2026 in IJEAST (http://www.ijeast.com)

The expressive power of a model also depends strongly on how data is encoded [14], and hardware noise reduces the information that can be read out. Many published studies report results on one dataset, with one circuit setting, and without comparing against simple classical baselines run on the same reduced features. This makes it hard to tell whether a reported gain comes from the quantum part or from the preprocessing.

In this paper we build a hybrid framework and test it under controlled conditions. Our contributions are as follows:

- A complete pipeline in which classical standardization, Principal Component Analysis (PCA) and angle scaling prepare the data, a variational quantum circuit (VQC) produces the representation, and the Adam optimizer trains the circuit parameters.
- A comparison against Logistic Regression (LR), an RBFkernel SVM and a Multi-Layer Perceptron (MLP), all trained on the same PCA features, across three benchmark datasets and three random splits.
- A study of circuit depth, number of qubits (retained features) and depolarizing noise, with training time reported for each setting.

The rest of the paper is arranged as follows. Section II describes the framework and its mathematical model. Section III gives implementation details. Section IV reports the simulation results, and Section V concludes the work.

II. METHODOLOGY

The framework has four stages: classical preprocessing, quantum data encoding, a trainable variational circuit, and a classical measurement-to-label head with an optimizer. Fig. 1 shows the circuit used in the quantum stage.

A. Classical Preprocessing

Let the dataset be $D = \{(x_i, y_i)\}_{i=1}^{N}$ with $x_i \in R^d$ and $y_i \in \{0, \ldots, C-1\}$. Each feature is first standardized using the mean μ_j and standard deviation σ_j computed on the training split only:

$$\tilde{x}_{ij} = \frac{x_{ij} - \mu_j}{\sigma_j}. \qquad (1)$$

Since the number of qubits n is far smaller than d for most real data, PCA projects the standardized vector onto the n leading eigenvectors $W_n \in R^{d\times n}$ of the training covariance matrix:

$$\mathbf{z}_i = \mathbf{W}_n^{\top}\tilde{\mathbf{x}}_i, \qquad \eta_n = \frac{\sum_{k=1}^{n}\lambda_k}{\sum_{k=1}^{d}\lambda_k}, \qquad (2)$$

where λ_k are the eigen values and η_n is the fraction of variance kept. This step is our feature selection mechanism. Finally, every component is rescaled to the interval $[0, \pi]$ so that it can be used directly as a rotation angle:

$$\phi_{ij} = \pi\,\frac{z_{ij} - z_j^{\min}}{z_j^{\max} - z_j^{\min}}. \qquad (3)$$

Test values falling outside the training range are clipped to $[0, \pi]$.

B. Quantum Data Encoding

We use angle encoding, in which feature j controls a Y - rotation on qubit j. Starting from $|0\rangle^{\otimes n}$, the encoded state is

$$|\psi(\boldsymbol{\phi})\rangle = \bigotimes_{j=1}^{n} R_Y(\phi_j)|0\rangle = \bigotimes_{j=1}^{n}\left(\cos\tfrac{\phi_j}{2}|0\rangle + \sin\tfrac{\phi_j}{2}|1\rangle\right). \qquad (4)$$

This encoding needs only one gate per feature and so keeps the circuit shallow, which matters on noisy devices. Its drawback is that the number of qubits grows linearly with the number of features, which is why the PCA step above is necessary.

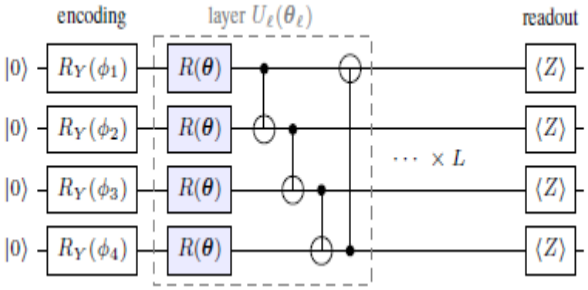


Fig. 1: Variational circuit for n = 4 qubits: angle encoding, L repeated layers of general rotations and a CNOT ring, and Pauli-Z readout.

C. Variational Quantum Circuit

After encoding, L trainable layers are applied. Each layer has a general single-qubit rotation on every qubit followed by a ring of CNOT gates that entangles neighbouring qubits:

International Journal of Engineering Applied Sciences and Technology, 2026
Vol. 11, Issue 06, ISSN No. 2455-2143, Pages 30-36
Published Online October 2026 in IJEAST (http://www.ijeast.com)

$$R(\alpha,\beta,\gamma) = R_Z(\gamma)\, R_Y(\beta)\, R_Z(\alpha), \tag{5}$$

$$U_\ell(\boldsymbol{\theta}_\ell) = \left(\prod_{j=1}^{n} \mathrm{CNOT}_{j,(j \bmod n)+1} \right) \times \bigotimes_{j=1}^{n} R(\boldsymbol{\theta}_{\ell,j}). \tag{6}$$

The complete circuit and the output state are

$$|\Psi(\phi, \theta)\rangle = U_L(\theta_L) \cdots U_1(\theta_1)|\psi(\phi)\rangle. \tag{7}$$

The circuit therefore has 3nL trainable angles. For n = 4 and L = 3 this is only 36 parameters.

D. Measurement and Classical Head

The quantum stage returns Pauli-Z expectation values

$$\langle Z_k \rangle = \langle \Psi(\phi, \theta)|Z_k|\Psi(\phi, \theta)\rangle \in [-1, 1]. \tag{8}$$

Because these values are bounded, a trainable scale s and bias b convert them into logits. For binary tasks only the first qubit is read:

$$\hat{p}(y = 1 \mid \mathbf{x}) = \sigma\big(s\langle Z_1 \rangle + b\big), \qquad \sigma(u) = \frac{1}{1 + e^{-u}}. \tag{9}$$

For C > 2 classes, qubit k provides the score of class k and a softmax gives

$$\hat{p}(y = k \mid \mathbf{x}) = \frac{\exp\left(s\langle Z_k \rangle + b_k\right)}{\sum_{c=1}^{C} \exp\left(s\langle Z_c \rangle + b_c\right)}. \tag{10}$$

E. Loss Function and Optimization

The model is trained with cross-entropy loss over mini batches B:

$$\mathcal{L}(\boldsymbol{\theta}, s, \mathbf{b}) = -\frac{1}{|\mathcal{B}|} \sum_{i \in \mathcal{B}} \sum_{k=1}^{C} \mathbf{1}[y_i = k] \times \log \hat{p}(y = k \mid \mathbf{x}_i). \tag{11}$$

Algorithm 1 Training of the Hybrid Quantum-Classical Classifier

Input Data: Training set $\mathcal{D}$, qubits n, layers L, epochs E, batch size B, learning rate η
Output: Trained parameters $\boldsymbol{\theta}, s, \mathbf{b}$

```
Fit scaler, PCA(n) and [0, π] scaler on D; transform inputs to φ_i
Initialize θ ~ U(0, 2π), s ← 2, b ← 0
for e = 1 to E do
    Shuffle D and split into batches of size B
    for each batch B do
        Encode φ_i using Eq. (4); apply U_L ··· U_1
        Measure ⟨Z_k⟩ and compute p̂ using Eq. (9) or (10)
        Compute L and gradients; update θ, s, b with Adam
    end for
end for
return θ, s, b
```

On real hardware, gradients of expectation values with respect to rotation angles can be computed exactly with the parameter shift rule [8], [15]:

$$\frac{\partial \langle Z_k \rangle}{\partial \theta} = \frac{1}{2}\left[\langle Z_k \rangle_{\theta + \frac{\pi}{2}} - \langle Z_k \rangle_{\theta - \frac{\pi}{2}}\right]. \tag{12}$$

In our simulator, the same gradients are obtained by automatic differentiation, which gives identical values at lower cost. The parameters are updated with Adam [16]:

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \eta \frac{\hat{\mathbf{m}}_t}{\sqrt{\hat{\mathbf{v}}_t} + \epsilon}, \tag{13}$$

where $\hat{m}_t$ and $\hat{v}_t$ are the bias-corrected first and second moment estimates of the gradient. The full training procedure is summarized in Algorithm 1.

F. Noise Model

To estimate how the trained model would behave on im perfect hardware, a depolarizing channel is inserted on every qubit after each variational layer:

$$\mathcal{E}(\rho) = (1 - p)\,\rho + \frac{p}{3}\left(X\rho X + Y\rho Y + Z\rho Z\right), \tag{14}$$

where p is the error probability per qubit per layer. The noisy circuit is simulated with density matrices. In this study the noise is applied at inference time to models trained without noise, which matches the common situation in

International Journal of Engineering Applied Sciences and Technology, 2026
Vol. 11, Issue 06, ISSN No. 2455-2143, Pages 30-36
Published Online October 2026 in IJEAST (http://www.ijeast.com)

which a model is trained in simulation and later deployed on a device.

G. Evaluation Metrics

With T P, T N, F P and F N counted per class, we report

$$\text{Acc} = \frac{TP + TN}{TP + TN + FP + FN}, \tag{15}$$

$$P = \frac{TP}{TP + FP}, \qquad R = \frac{TP}{TP + FN},$$

$$F_1 = \frac{2PR}{P + R}. \tag{16}$$

TABLE I: Benchmark Datasets Used in the Experiments

Dataset	Samples	Features	Classes	Train/Test
Iris	150	4	3	105/45
Wine	178	13	3	124/54
Breast ancer	569	30	2	398/171

Precision, recall and F_1 are macro-averaged over classes so that small classes count equally. Computational cost is measured as the number of trainable parameters and the wall clock training time.

III. IMPLEMENTATION AND EXPERIMENTAL SETUP

A. Software and Hardware

The framework was written in Python 3.12. The quantum part uses PennyLane 0.45 [17] with the default.qubit state-vector simulator for noiseless runs and default.mixed for noisy runs. Preprocessing and the classical baselines use scikit-learn 1.8 [18]. The same circuit can be exported to Qiskit [19] for execution on IBM hardware, although all results here come from simulation. All experiments ran on a single core of an Intel Xeon processor at 2.10 GHz with 3 GB of RAM and no GPU.

B. Datasets

Three standard datasets from the UCI repository, distributed with scikit-learn, were used. They are listed in Table I. Each dataset was split into 70% training and 30% testing with class stratification, and every experiment was repeated with three random seeds (0, 1, 2). We report the mean and standard deviation over these runs.

C. Model Configuration

Unless stated otherwise, the VQC uses n = 4 qubits and L = 3 layers, giving 36 circuit angles plus a scale and bias terms (38 parameters for the binary task and 40 for the three class tasks). Training ran for 40 epochs with batch size 16 and learning rate 0.05. All baselines receive exactly the same four PCA features. LR uses L2 regularization with C = 1. The SVM uses an RBF kernel with C = 1 and γ set to scale [20]. The MLP has two hidden layers of 16 and 8 ReLU units trained with Adam, which gives 225 parameters for the binary task. Giving every model the same input isolates the effect of the learner itself from the effect of dimensionality reduction.

IV. RESULTS AND DISCUSSION

A. Comparison with Classical Models

Table II shows the main results. On Iris the hybrid model gave the highest mean accuracy, 98.52%, ahead of the MLP at 97.78% and LR and SVM at 97.04%. It also had the lowest spread across seeds. On Breast Cancer the VQC reached 94.93%, which lies between LR (94.74%) and SVM/MLP (95.13%). On Wine it reached 93.21%, above the MLP but below SVM (95.06%) and LR (94.44%).

TABLE II: Classification Performance on Test Sets (Mean ±Std over 3 Seeds, 4 PCA Features, Values in %)

Data	Model	Acc.	Prec.	Rec.	F1	Time (s)
Iris	LR	97.04±2.77	97.12	97.04	97.03	<0.01
	SVM	97.04±2.10	97.39	97.04	97.02	<0.01
	MLP	97.78±1.81	98.00	97.78	97.77	0.17
	VQC	**98.52±1.05**	**98.61**	**98.52**	**98.52**	**7.75**
Wine	LR	94.44±1.51	94.94	94.97	94.74	<0.01
	SVM	**95.06±0.87**	**95.50**	**95.50**	**95.42**	<0.01
	MLP	92.59±0.00	93.16	93.21	93.03	0.15
	VQC	93.21±0.87	93.80	93.62	93.52	8.93
Breast Cancer	LR	94.74±0.95	**95.06**	93.70	94.30	<0.01
	SVM	**95.13±1.46**	94.95	**94.64**	**94.78**	<0.01
	MLP	**95.13±0.99**	94.94	**94.64**	**94.78**	0.17
	VQC	94.93±1.68	94.98	94.17	94.54	23.92

International Journal of Engineering Applied Sciences and Technology, 2026
Vol. 11, Issue 06, ISSN No. 2455-2143, Pages 30-36
Published Online October 2026 in IJEAST (http://www.ijeast.com)

Two points should be stated clearly. First, the gaps between models are smaller than one standard deviation in almost every case. With test sets of 45 to 171 samples, a difference of one or two misclassified points moves accuracy by about 0.6– 2.2%. We therefore read Table II as showing that the hybrid model is competitive with classical learners, not that it is better. Second, the hybrid model needs far more compute. Training in simulation took between 7.75 s and 23.92 s, while LR and SVM finished in milliseconds. This overhead comes from simulating the quantum state, and it grows with the size of the training set, which is why Breast Cancer took about three times longer than the other two datasets.

The Wine result is also informative. Four principal com ponents keep only 75.0% of the variance of its 13 features, compared with 100% for Iris. All four models lost accuracy relative to what is usually reported on the full feature set, and the drop says more about the qubit budget than about the learner. In practice, the number of available qubits sets an upper limit on how much information reaches the classifier.

B. Training Convergence

Fig. 2 shows the training loss per epoch, averaged over seeds. Loss drops quickly in the first ten epochs on all three datasets, and we did not see a barren plateau at this scale. The Breast Cancer curve is noisier after epoch 15. This is a mini-batch effect: with 398 training samples and batch size 16, each epoch contains 25 updates at a fixed learning rate of 0.05, so the optimizer keeps moving around the minimum. A decaying learning rate would likely smooth the curve.

C. Effect of Circuit Depth

Table III varies the number of layers on Breast Cancer with n = 4. A single layer underfits badly. Accuracy is 73.88% and the final loss stays at 0.606, close to that of a constant predictor. With one layer, each qubit passes through only a single CNOT ring, so the readout qubit sees very little of the other features. Moving to two layers lifts accuracy to 93.57%, and three layers give the best result of 94.93%. Beyond that, accuracy falls slightly to 94.35% and the final training loss rises. Extra layers make the landscape harder to optimize 1 12 73.88±3.11 69.47 0.6057 9.26 2 24 93.57±0.83 93.05 0.0859 17.44 3 36 94.93±1.68 94.54 0.0730 25.31 4 48 94.74±0.95 94.36 0.0770 34.58 5 60 94.35±1.20 93.98 0.0893 42.83 6 72 94.35±1.20 94.01 0.0913 49.18 within a fixed budget of 40 epochs, which is consistent with the trend predicted by barren plateau analysis [13]. Training time, measured in separate sequential runs, grows almost linearly at about 8 s per added layer.

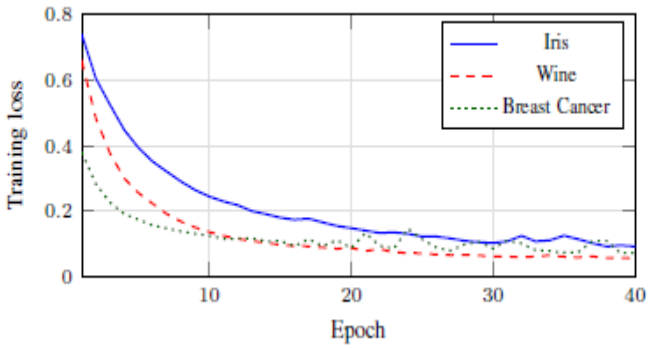


Fig. 2: Mean training loss of the VQC (n = 4, L = 3) over 40 epochs.

TABLE III: Effect of Circuit Depth L (Breast Cancer, n = 4)

L	Params	Acc. (%)	F1 (%)	Final loss	Time (s)
1	12	73.88±3.11	69.47	0.6057	9.26
2	24	93.57±0.83	93.05	0.0859	17.44
3	36	**94.93±1.68**	**94.54**	**0.0730**	25.31
4	48	94.74±0.95	94.36	0.0770	34.58
5	60	94.35±1.20	93.98	0.0893	42.83
6	72	94.35±1.20	94.01	0.0913	49.18

D. Effect of Feature Selection and Qubit Count

Since one qubit carries one PCA component, the qubit count and the number of selected features are the same setting. Table IV shows the results on Breast Cancer with L = 3. With only two components (63.8% of variance) the model already reaches 93.57%, which shows that this dataset is close to linearly separable along its leading directions. Accuracy rises to 95.52% at six qubits, where 88.8% of the variance is retained. The trend is not strictly monotonic, as five qubits score slightly below four, but this difference is within the seed variation. The cost side is more predictable. Each added qubit doubles the size of the simulated state vector, and training time rose from 12.5 s to 38.4 s across the range tested. On hardware, the cost would instead appear as wider circuits and more two-qubit gates, each of which adds error.

International Journal of Engineering Applied Sciences and Technology, 2026
Vol. 11, Issue 06, ISSN No. 2455-2143, Pages 30-36
Published Online October 2026 in IJEAST (http://www.ijeast.com)

E. Effect of Simulated Quantum Noise

Fig. 3 and Table V show what happens when depolarizing noise is added to the trained Breast Cancer model. Up to p = 0.02 per qubit per layer, test accuracy does not change at all. At p = 0.05 and p = 0.10 the drop is only 0.2–0.4 percentage points. At p = 0.20 accuracy falls sharply to 89.28% and the spread across seeds grows to 2.63%.

The reason for this tolerance is fairly simple. Depolarizing noise pulls each expectation value $\langle Z_1 \rangle$ towards zero but rarely changes its sign. For a binary classifier whose decision depends on the sign of $s\langle Z_1 \rangle + b$, the shrinkage mostly affects samples that were already close to the boundary. Once p

TABLE IV: Effect of Qubit Count / Retained PCA Features (Breast Cancer, L = 3)

n	Var. kept	Params	Acc. (%)	F1 (%)	Time (s)
2	63.8%	18	93.57±1.72	93.17	12.52
3	72.8%	27	93.76±1.46	93.29	16.42
4	79.4%	36	94.93±1.68	94.54	25.31
5	84.6%	45	94.35±0.99	93.96	31.33
6	88.8%	54	**95.52±0.55**	**95.21**	38.35

TABLE V: Test Performance under Depolarizing Noise (Breast Cancer, n = 4, L = 3)

Noise p	Acc. (%)	F1 (%)	ΔAcc. (pts)
0.00	94.93±1.68	94.54±1.80	–
0.01	94.93±1.53	94.56±1.64	0.00
0.02	94.93±1.53	94.56±1.64	0.00
0.05	94.74±1.26	94.34±1.34	−0.19
0.10	94.54±1.20	94.12±1.30	−0.39
0.20	89.28±2.63	88.29±3.09	−5.65

becomes large, the shrunken outputs get dominated by the bias term b, and borderline samples start to switch class. Two qualifications apply. Probabilities and confidence scores degrade much earlier than hard labels do, so any application that relies on calibrated outputs would be affected at lower p. Also, the simple channel used here leaves out readout error, crosstalk and finite-shot sampling, all of which are present on real devices.

F. Discussion and Limitations

Taken together, the results show where a small hybrid model stands today. With the same features, a 4-qubit, 3-layer circuit with about 40 parameters matched classical models that are cheaper to train, including an MLP with over five times more parameters. Hybrid learning was most competitive when the reduced input kept most of the original information (Iris), when the circuit had two to four layers, and when gate noise stayed at or below about 10% per layer. It fell behind when PCA discarded a large share of the variance (Wine). In none of the settings did it give a statistically clear advantage, and training was between roughly 50 and several thousand times slower in simulation. These findings agree with the view that, for small classical datasets, near-term quantum models should be judged by parameter efficiency and noise behaviour rather than by raw accuracy [6], [12].

The study has limits that should be kept in mind. The datasets are small and well studied, three seeds give only a rough estimate of variance, and all runs used a simulator rather than hardware. Noise was applied only during inference. Future work will train the model under noise, use shot-based estimation, run on IBM devices through Qiskit, and test data re-uploading encodings that may allow fewer qubits to carry more features [14].

V. CONCLUSION

This paper presented a hybrid quantum-classical classifi cation framework in which classical preprocessing prepares compact features, a variational quantum circuit learns the representation, and a classical optimizer trains it. Experiments on Iris, Wine and Breast Cancer showed that the hybrid model reached 98.52%, 93.21% and 94.93% test accuracy, within one standard deviation of Logistic Regression, SVM and MLP baselines trained on the same features. Circuit depth had the largest effect: one layer underfit, while three layers gave the best accuracy, and deeper circuits brought no further gain. Increasing the number of qubits improved accuracy slowly while increasing cost steadily. The trained model kept its accuracy under depolarizing noise up to p = 0.10 but lost more than five points at p = 0.20. The main practical barriers remain the limited qubit count, which forces aggressive feature reduction, and the high cost of simulation. Hybrid models are therefore a reasonable research tool and a parameter-efficient option, but they do not yet replace classical classifiers on standard tabular problems.

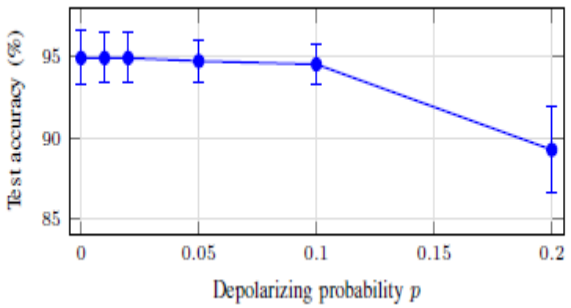


Fig. 3: VQC test accuracy versus depolarizing noise level (mean ± std over 3 seeds).

VI. REFERENCES

[1]. Preskill J. (2018). Quantum Computing in the NISQ Era and Beyond, Quantum, vol. 2, (pp.79), DOI: 10.22331/q-2018-08-06-79.

[2]. Biamonte J., Wittek P., Pancotti N., Rebentrost P., Wiebe N., and Lloyd S. (2017). Quantum Machine Learning, Nature, vol. 549, (pp.195-202), DOI: 10.1038/nature23474.

[3]. Havl´ıcek V., C ˇ orcoles A. D., Temme K., Harrow A. W., Kandala A., ´ Chow J. M., and Gambetta J. M. (2019). Supervised Learning with Quantum-Enhanced Feature Spaces, Nature, vol. 567, (pp.209-212), DOI: 10.1038/s41586-019-0980-2.

[4]. Schuld M., and Killoran N. (2019). Quantum Machine Learning in Feature Hilbert Spaces, Physical Review Letters, vol. 122, no. 4, (Art. 040504), DOI: 10.1103/PhysRevLett.122.040504.

[5]. Cerezo M., Arrasmith A., Babbush R., Benjamin S. C., Endo S., Fujii K., McClean J. R., Mitarai K., Yuan X., Cincio L., and Coles P. J. (2021). Variational Quantum Algorithms, Nature Reviews Physics, vol. 3, (pp.625-644), DOI: 10.1038/s42254-021-00348-9.

[6]. Schuld M., and Petruccione F. (2021). Machine Learning with Quantum Computers, 2nd ed., Springer, Cham, DOI: 10.1007/978-3-030-83098-4.

[7]. Farhi E., and Neven H. (2018). Classification with Quantum Neural Networks on Near Term Processors, arXiv preprint arXiv:1802.06002, DOI: 10.48550/arXiv.1802.06002.

[8]. Mitarai K., Negoro M., Kitagawa M., and Fujii K. (2018). Quantum Circuit Learning, Physical Review A, vol. 98, no. 3, (Art. 032309), DOI: 10.1103/PhysRevA.98.032309.

[9]. Schuld M., Bocharov A., Svore K. M., and Wiebe N. (2020). Circuit Centric Quantum Classifiers, Physical Review A, vol. 101, no. 3, (Art. 032308), DOI: 10.1103/PhysRevA.101.032308.

[10]. Benedetti M., Lloyd E., Sack S., and Fiorentini M. (2019). Parameterized Quantum Circuits as Machine Learning Models, Quantum Science and Technology, vol. 4, no. 4, (Art. 043001), DOI: 10.1088/2058- 9565/ab4eb5.

[11]. Mari A., Bromley T. R., Izaac J., Schuld M., and Killoran N. (2020). Transfer Learning in Hybrid Classical-Quantum Neural Networks, Quan tum, vol. 4, (pp.340), DOI: 10.22331/q-2020-10-09-340.

[12]. Abbas A., Sutter D., Zoufal C., Lucchi A., Figalli A., and Woerner S. (2021). The Power of Quantum Neural Networks, Nature Computational Science, vol. 1, (pp.403-409), DOI: 10.1038/s43588-021-00084-1.

[13]. McClean J. R., Boixo S., Smelyanskiy V. N., Babbush R., and Neven H. (2018). Barren Plateaus in Quantum Neural Network Train ing Landscapes, Nature Communications, vol. 9, (Art. 4812), DOI: 10.1038/s41467-018-07090-4.

[14]. Schuld M., Sweke R., and Meyer J. J. (2021). Effect of Data Encoding on the Expressive Power of Variational Quantum Machine-Learning Models, Physical Review A, vol. 103, no. 3, (Art. 032430), DOI: 10.1103/PhysRevA.103.032430.

[15]. Schuld M., Bergholm V., Gogolin C., Izaac J., and Killoran N. (2019). Evaluating Analytic Gradients on Quantum Hardware, Physical Review A, vol. 99, no. 3, (Art. 032331), DOI: 10.1103/PhysRevA.99.032331.

[16]. Kingma D. P., and Ba J. (2015). Adam: A Method for Stochastic Optimization, in Proc. 3rd International Conference on Learning Repre sentations (ICLR), San Diego, USA, DOI: 10.48550/arXiv.1412.6980.

[17]. Bergholm V., Izaac J., Schuld M., Gogolin C., Ahmed S., Ajith V., et al. (2018). PennyLane: Automatic Differentiation of Hybrid Quantum-Classical Computations, arXiv preprint arXiv:1811.04968, DOI: 10.48550/arXiv.1811.04968.

[18]. Pedregosa F., Varoquaux G., Gramfort A., Michel V., Thirion B., Grisel O., et al. (2011). Scikit-learn: Machine Learning in Python, Journal of Machine Learning Research, vol. 12, (pp.2825-2830).

[19]. Javadi-Abhari A., Treinish M., Krsulich K., Wood C. J., Lishman J., Gacon J., et al. (2024). Quantum Computing with Qiskit, arXiv preprint arXiv:2405.08810, DOI: 10.48550/arXiv.2405.08810.

[20]. Cortes C., and Vapnik V. (1995). Support-Vector Networks, Machine Learning, vol. 20, no. 3, (pp.273-297), DOI: 10.1007/BF00994018.

IJEAST

INTERNATIONAL JOURNAL OF ENGINEERING APPLIED SCIENCE AND TECHNOLOGY

AUTHORS

Shweta Narayan Bamne

Prof M.S. Shegokar

ABOUT IJEAST

International journal of Engineering Applied Science and Technology (IJEAST) is a peer-reviewed, open access journal that publishes high - quality research paper in the field of Engineering, Applied Science and Technology.

IJEAST aims to provide a platform for researchers, academicians, and professionals to share their innovative ideas, research findings, and practical experiences with the global scientific community.

JOURNAL METRICS

H-INDEX	i10-INDEX	IJEAST CITATIONS	PEER-REVIEWED & OPEN ACCESS
33	154	9,647 (IJEAST CITATIONS)	PUBLISHED MONTHLY

PEER REVIEWED

All submission are rigorously peer reviewed to ensure quality.

AUTHORS ARE OUR PRIORITY

We value our authors and recognize their contribution to the advancement of research and knowledge.

OPEN ACCESS

Free and unrestricted access to research for all.

GLOBAL REACH

Connecting researchers and professionals worldwide.

TIMELY PUBLICATION

We ensure a swift and efficient publication process.

For more information, visit our website
www.ijeast.com

editor@ijeast.com

www.ijeast.com

India

2455-2143