



Comparative Analysis of Quantum Machine Learning Algorithms: Variational Quantum Classifiers, Quantum Support Vector Machines, and Hybrid Quantum-Classical Networks

Jampana Usha Rani ¹ , Kondapalli Venkata Ramana ²

Department of Computer science & System Engineering, Andhra University College of engineering, Visakhapatnam, India ; 424206454008@andhrauniversity.edu.in , dr.kondapallivr@andhrauniversity.edu.in

* Corresponding Author : Jampana Usha Rani; 424206454008@andhrauniversity.edu.in

Abstract: Quantum machine learning (QML) aims to use quantum-in-the-knowledge resources to improve learning performances. How current out-of-the-box algorithms compare with carefully optimized classical algorithms is unclear at present. We compare QML paradigms all quantum models run on a noise-free state vector simulator (Qiskit Aer) against three classical baselines: a kernel SVM, Random Forest and XGBoost. The three QML models are (i) the Variational Quantum Classifier (VQC), (ii) a quantum-optimized support vector machine (QSVM) and (iii) a hybrid quantum-classical Multi-Layer Perceptron (HQ-MLP). Quantum models are trained on subsets of the binary MNIST dataset (digits 0 vs. 1, PCA-reduced to between 4 and 8 features) and tested on the remaining data. In addition, we use four synthetic, two-class benchmark problems: circles, moons, a multi-cluster data set, and spirals. Accuracy, the area under the ROC curve (AUC), training time and prediction time are reported for each. On the MNIST dataset, the quantum and hybrid models reach 97.50–98.33% accuracy, within one percentage point of the classical models (99.33%). On the synthetic data sets, quantum and hybrid models perform similarly (94.00–96.25%) to classical SVM and XGBoost (92.50%) but less well than Random Forest (100%). The HQ-MLP attains the highest accuracy among the quantum models at near-classical cost, with VQC and QSVM requiring approximately one to two orders of magnitude longer to train and more than an order of magnitude longer to infer at in. Simulation. We explain why these outcomes should not be taken as evidence for quantum advantage, examine threats to validity and enumerate the experiments needed to demonstrate it.

Keywords: QML, VQC, quantum support vector machine, QSVM, NISQ, MNIST, benchmarking.

1. Introduction

Quantum Machine Learning after all, is simply the search for structure in high-dimensional data. The quantum computer provides another computational substrate to perform that search: an n -qubit register is in a state space of dimension 2^n , and quantum primitives like superposition, interference, and entanglement can, in principle, manipulate that space in ways that are prohibitively expensive for classical hardware [1, 2]. Early quantum algorithms for linear algebra and learning showed polynomial or even exponential speedups for fundamental problems like least-squares fitting and support vector classification [5, 4, 3]. Unfortunately, those speedups relied on assumptions about data loading, state preparation, and readout that are difficult to meet in

practice [6], and several of those algorithms have since been matched by “dequantized” classical algorithms [7].

The devices used today are the Noisy Intermediate-Scale Quantum (NISQ) generation [8] : tens to hundreds of noisy qubits, with no error correction. Algorithms designed for this regime are consequently shallow, hybrid, variational. A parametrized quantum circuit runs on the device and is optimized by a classical optimizer [9, 10, 11]. Two classes of quantum models have been developed along these principles: variational (“quantum neural network”) classifiers [12, 13, 14], and quantum kernel methods which use Consider a quantum circuit as a feature map, and combine it with a classical support vector machine to perform optimization [18, 19]. The third, pragmatic family,



include a quantum layer inside a classical neural network [23]. Even with this considerable theoretical body of work, we still lack good side-by-side empirical comparisons of these families on common data with common metrics, and recent work has warned that sloppy benchmarking can readily inflate the evidence in favor of quantum models [24]. Here, our aim is to present such a comparison transparently, and to be clear on what it can and cannot show.

1.1. Contributions

In this paper, we make the following contributions.

- **Unified Approach:** We introduce a VQC, a QSVM, and an HQ-MLP, which share a common data pipeline, simulation environment, and evaluation protocol. We compare these to an SVM, Random Forest, and XGBoost baseline.
- **Multi-Metric Evaluation:** Accuracy, AUC, training time, and prediction time are measured on binary MNIST and four synthetic geometries. The results are aggregated by model family in Tables IV–VII.
- **Cost Comparison:** We measure the simulation overhead of each quantum model against the classical baselines, and compare this to the structure of each algorithm (Section V-E).
- **Critical Evaluation:** We interpret the implications of the results for quantum advantage, enumerate validity threats and describe the experiments needed for a more conclusive result (Sections VI and VII).

1.2. Paper Organization

Section II covers related work. Section III contains the background of the three models. Section IV contains the datasets, implementations, and evaluation protocol. Section V the results, Section VI the discussion, Section VII analyses threats to validity, and Section IX concludes.

2. Related Work

Quantum Algorithms For Learning: Lloyd, Mohseni and Rebentrost proposed quantum algorithms for supervised and unsupervised learning based on quantum linear algebra [3], and Rebentrost et al. gave a quantum SVM with a runtime logarithmic in the data dimension assuming efficient quantum random access memory [4]. These are based on the HHL algorithm for linear systems [5], although Aaronson’s analysis of the “fine print” of such claims loading the input, maintaining the condition number, and extracting the output remains an important caveat [6], and Tang showed that for the recommendation-systems problem a classical algorithm can match the quantum one up to polynomial factors [7].

Variational Models: An alternative to the use of fault-tolerant subroutines in the NISQ era is to replace these components with trainable circuits. Classification with

quantum neural networks on near-term quantum processors was proposed by Farhi and Neven [12]; quantum circuit learning was introduced by Mitarai et al [13]; and circuit-centric quantum classifiers with a specifically shallow architecture were proposed by Schuld et al [14]. The expectation value gradients are computable on hardware via the parameter-shift rule [15]. Cerezo et al and Bharti et al provide surveys of variational algorithms and their issues [10, 11], including barren plateaus where the gradients decay exponentially with the number of qubits in the case of sufficiently random circuits [16]. Abbas et al. analysed the expressive capacity of quantum neural networks through their effective dimension [17].

Quantum Kernels: Havlíček et al. Demonstrated a quantum-enhanced feature map and kernel estimation using a superconducting processor [18], while Schuld and Killoran established that both methods can be interpreted as operating within a feature Hilbert space [19]. Later studies examined the inductive bias present in quantum kernels [20], investigated how the properties of the data influence whether a quantum model can surpass a classical one [21], and explored the training of embedding kernels on near-term devices [22].

The Hybrid Approach: Mari et al. Looked at transfer learning in classical-quantum hybrid neural networks, which is where a classical network has been pretrained before it’s connected to a small variational quantum circuit [23]. The idea is that there’s a small quantum component, but it’s hard to compare its benefits to just having the same size classical layers.

Benchmarking. Bowles, Ahmed, and Schuld recently benchmarked a huge dataset of quantum classifiers versus classical baselines and found that, straightforwardly, the classical models usually do better, and that getting rid of entanglement usually doesn’t make it worse [24]. Ours is more modest, but in the same spirit: we include strong classical baselines, and aim to test advantage as a hypothesis, not an expectation.

3. Theoretical Background

3.1. Quantum States and Circuits

A single qubit is described by a normalized vector in $\mathbb{C}^2$,

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle, \quad |\alpha|^2 + |\beta|^2 = 1. \quad (1)$$

An n -qubit state lives in a 2^n -dimensional Hilbert space. A quantum circuit is a sequence of unitary gates; the single-qubit rotations used in this work are

$$R_P(\vartheta) = e^{-i\vartheta P/2}, \quad P \in \{X, Y, Z\}, \quad (2)$$

and entanglement is generated by two-qubit gates such as CNOT and $R_{ZZ}(\vartheta) = e^{-i\vartheta Z \otimes Z/2}$. A measurement of an observable O returns, in expectation, $\langle\psi|O|\psi\rangle$. On a statevector simulator this expectation is computed exactly; on hardware it is estimated from a finite number of shots.

3.2. Variational Quantum Classifier

A VQC maps an input $\mathbf{x} \in \mathbb{R}^d$ to a quantum state by an encoding circuit $S(\mathbf{x})$, applies a trainable unitary $U(\boldsymbol{\theta})$, and reads out an observable:

$$|\psi(\mathbf{x}, \boldsymbol{\theta})\rangle = U(\boldsymbol{\theta}) S(\mathbf{x}) |0\rangle^{\otimes n}, \quad (3)$$

$$f(\mathbf{x}; \boldsymbol{\theta}) = \langle \psi(\mathbf{x}, \boldsymbol{\theta}) | Z_1 | \psi(\mathbf{x}, \boldsymbol{\theta}) \rangle. \quad (4)$$

The predicted label is $\hat{y} = \text{sign } f(\mathbf{x}; \boldsymbol{\theta})$. The parameters minimize an empirical loss over N training pairs,

$$\mathcal{L}(\boldsymbol{\theta}) = \frac{1}{N} \sum_{i=1}^N \ell(y_i, f(\mathbf{x}_i; \boldsymbol{\theta})), \quad (5)$$

using a classical optimizer. For gates of the form $R_P(\theta)$, the gradient of f can be obtained exactly from two shifted circuit evaluations [13, 15],

$$\frac{\partial f}{\partial \theta_k} = 1/2 [f(\boldsymbol{\theta} + \pi/2 \mathbf{e}_k) - f(\boldsymbol{\theta} - \pi/2 \mathbf{e}_k)], \quad (6)$$

so a full gradient costs $2P$ circuit evaluations per sample for P parameters. This is the main reason a VQC is expensive to train in simulation. For n qubits and L layers of the ansatz used here, $P = 2nL$.

3.3. Quantum Support Vector Machine

A QSVM uses a parameter-free feature map $U_\Phi(\mathbf{x})$ to embed data into Hilbert space, $|\phi(\mathbf{x})\rangle = U_\Phi(\mathbf{x})|0\rangle^{\otimes n}$, and estimates the kernel

$$K(\mathbf{x}_i, \mathbf{x}_j) = |\langle \phi(\mathbf{x}_i) | \phi(\mathbf{x}_j) \rangle|^2 = |\langle 0 | U_\Phi^\dagger(\mathbf{x}_j) U_\Phi(\mathbf{x}_i) | 0 \rangle|^2, \quad (7)$$

$U_\Phi(\mathbf{x}) \quad R_{ZZ} \text{ couplings} \quad U_\Phi^\dagger(\mathbf{x}')$

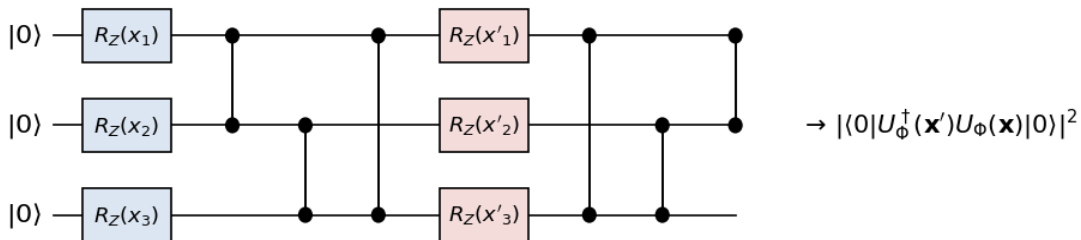


Fig. 1. Quantum kernel estimation circuit for three qubits (Eq. 7). The feature map $U_\Phi(\mathbf{x})$ (a layer of R_Z rotations followed by R_{ZZ} couplings, Eq. 9) is followed by the inverse map for the second data point; the kernel value is the probability of measuring $|0\rangle^{\otimes 3}$.

3.5. Summary of the Three Paradigms

Table I contrasts the three paradigms along the dimensions that matter for the cost and expressivity analysis in later sections.

Table.1 Comparison of the Three Quantum Paradigms

	VQC	QSVM	HQ-MLP
Trainable parameters	Circuit angles $\boldsymbol{\theta}, \boldsymbol{\phi}$ ($2nL$)	None in circuit; SVM dual variables α_i	Circuit (optional) and all MLP weights
Optimizer	Classical, over circuit parameters	Convex QP (SVM)	Gradient descent on MLP
Quantum workload	$\mathcal{O}(P)$ circuits per sample per gradient	$\mathcal{O}(N^2)$ kernel circuits	One circuit per sample per forward pass
Prediction	One circuit per sample	$\mathcal{O}(N_{sv})$ circuits per sample	One circuit + MLP
Main risk	Barren plateaus, local minima	Kernel concentration, cost of Gram matrix	Quantum layer may be redundant

as the probability of measuring the all-zeros string after applying $U_\Phi(\mathbf{x}_i)$ followed by $U_\Phi^\dagger(\mathbf{x}_j)$ (Fig. 1). The Gram matrix is then passed to a classical SVM, which solves the dual problem

$$\max_{\alpha} \sum_i \alpha_i - 1/2 \sum_{i,j} \alpha_i \alpha_j y_i y_j K_{ij}, \quad 0 \leq \alpha_i \leq C, \quad \sum_i \alpha_i y_i = 0. \quad (8)$$

The feature map used here follows the ZZ-type construction of Havlíček et al. [18],

$$U_\Phi(\mathbf{x}) = \prod_i R_Z(x_i) \prod_{i>j} R_{ZZ}(x_i x_j). \quad (9)$$

Training requires $\mathcal{O}(N^2)$ kernel evaluations and prediction requires $\mathcal{O}(N_{sv})$ evaluations per test point, where N_{sv} is the number of support vectors. These counts, rather than circuit depth, dominate the QSVM cost.

3.4. Hybrid Quantum-Classical Network

A hybrid network composes a quantum layer $Q: \mathbb{R}^d \rightarrow \mathbb{R}^m$ with classical layers. The quantum layer encodes the input, applies a fixed or trainable circuit, and returns expectation values $q_k = \langle Z_k \rangle$ as features $\mathbf{q} = (q_1, \dots, q_m)$. These are processed by a multi-layer perceptron with ReLU activations and a sigmoid output $\hat{y} = \sigma(\mathbf{w}^T h(\mathbf{q}) + b)$. Because most of the learnable capacity and most of the optimization sits in the classical layers, the quantum circuit is evaluated only as a feature extractor.

4. Research Methodology

4.1. Experimental Framework

The framework (Fig. 2) comprises a data stage, model stage (three quantum/hybrid models and three classical baselines), and evaluation stage. Quantum circuits were

built and simulated using Qiskit [25] with the Aer statevector backend, which produces theoretically exact expectation values, without shot or hardware noise. The classical baselines were implemented using scikit-learn [26] (SVM [27], Random Forest [28]) and XGBoost [29]. Details are summarized in Table II.

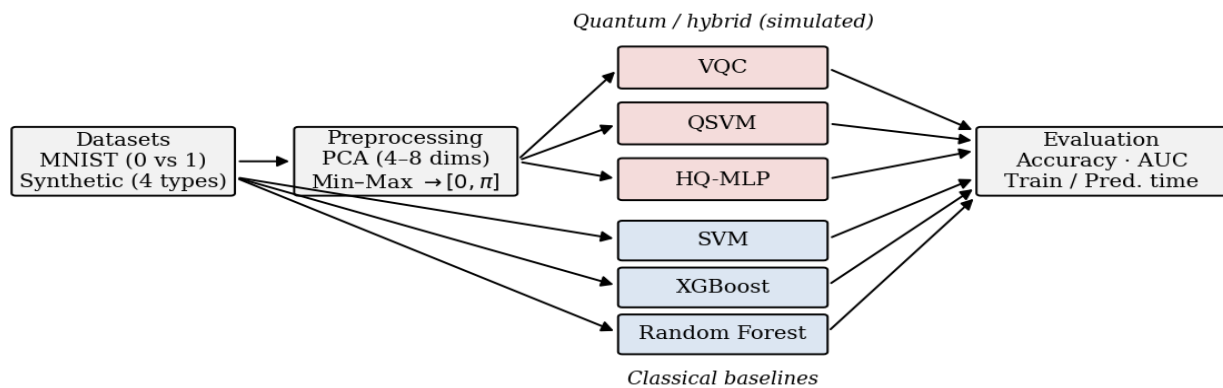


Fig. 2 Overview of the experimental framework. Quantum and hybrid models receive PCA-reduced, scaled features; all six models are evaluated with the same metrics.

Table. 2 Experimental Configuration

Item	Setting
Quantum simulator	Qiskit Aer, statevector backend (noise-free, exact expectations)
Qubits	4–8, equal to the PCA output dimension
Datasets	MNIST (digits 0 vs. 1); four synthetic two-class sets
Samples per experiment	600–800
Dimensionality reduction	PCA to 4–8 dimensions (quantum models)
Feature scaling	Min–Max to $[0, \pi]$ (angle encoding)
Quantum models	VQC, QSVM, HQ-MLP
Classical baselines	SVM, Random Forest, XGBoost
Metrics	Accuracy, AUC, training time, prediction time

4.2. Datasets

MNIST

MNIST [30] consists of 28×28 grey-scale images of handwritten digits. We consider the binary task of classifying the images into 0 or 1, subsampling 600–800 images for each experiment. The images are projected to 4–8 principal components with PCA in order to reduce the input dimensions in line with the increasing number of qubits; the extracted features are then Min–Max scaled to $[0, \pi]$ to be used directly as rotation angles. The 0-vs-1 task is arguably the easiest in MNIST; good accuracy can be reached by basically any learning model, and we show it

here as a sanity check on our pipeline instead of as a benchmarking tool.

Synthetic Datasets

4 two-dimensional synthetic tasks (Fig. 3) test different decision-boundary shapes: Circles (concentric rings, non-linearly separable), Moons (two off-set half-circles), Complex (multiple clusters per class), Spiral (two intertwined arms). Here we show their characteristics in Table III. The rows labelled “Synthetic” in the combined results are the average across all 4 synthetic tasks.

Table. 3 Datasets Used in the Study

Dataset	Type / input	Characteristic
MNIST (0 vs. 1)	Image; PCA to 4–8 d	Real data, nearly linearly separable
Circles	2-D synthetic	Radially separable; no linear boundary
Moons	2-D synthetic	Curved, interleaved boundary
Complex	2-D synthetic	Multi-cluster classes
Spiral	2-D synthetic	Highly non-linear, long boundary

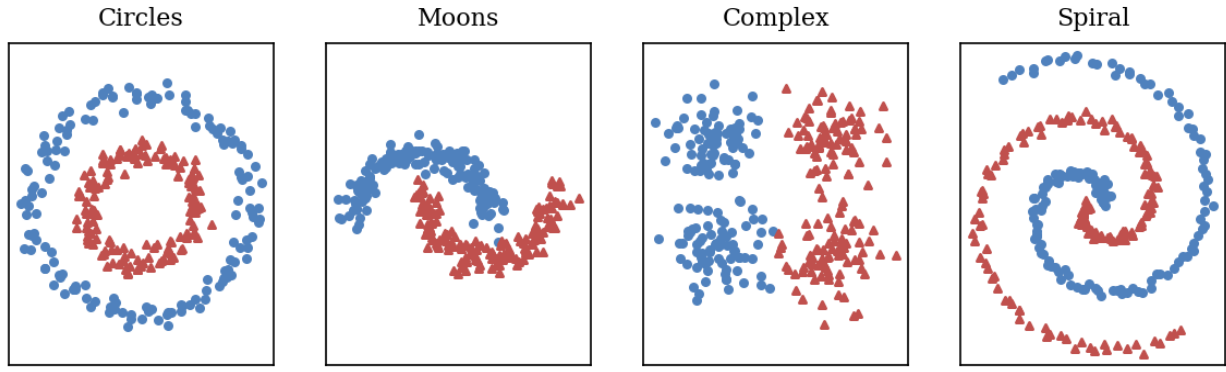


Fig. 3 Illustrative samples of the four synthetic problem types (generated for visualization; noise levels are indicative).

4.3. Model Implementations

4.3.1. Variational Quantum Classifier

The VQC (Fig. 4, Algorithm 1) encodes each feature on one qubit with an R_Y rotation, applies L layers of trainable $R_Y(\theta_{l,i})$ and $R_Z(\phi_{l,i})$ rotations on each qubit followed by a ring of CNOT gates, and measures $\langle Z \rangle$ on the first qubit. The classification decision is the sign of this expectation.

Algorithm 1: VQC forward pass for input $\mathbf{x}$

Step - 1 : Initialize n qubits in $|0\rangle$.

Step - 2 : Encode: apply $R_Y(\tilde{x}_i)$ to qubit i , for $i = 1, \dots, n$.

Step - 3 : For $l = 1$ to L : for $i = 1$ to n , apply $R_Y(\theta_{l,i})$ and $R_Z(\phi_{l,i})$ to qubit i ; then apply CNOT gates $(1,2), (2,3), \dots, (n,1)$.

Step - 4 : Measure $\langle Z_1 \rangle$ and predict $\hat{y} = \text{sign}\langle Z_1 \rangle$.

4.3.2. Quantum Support Vector Machine

The QSVM uses the ZZ-type feature map of Eq. (9). The Gram matrix is computed by the state vector simulator for all training pairs, and a precomputed-kernel SVM is trained on it. Test points are classified with the kernel values between each test point and the training set.

4.3.3. Hybrid Quantum-Classical MLP

The HQ-MLP (Fig. 5) passes the encoded input through a quantum layer that returns measurement-derived features, then through classical dense layers with ReLU activation, and a sigmoid output unit for binary classification.

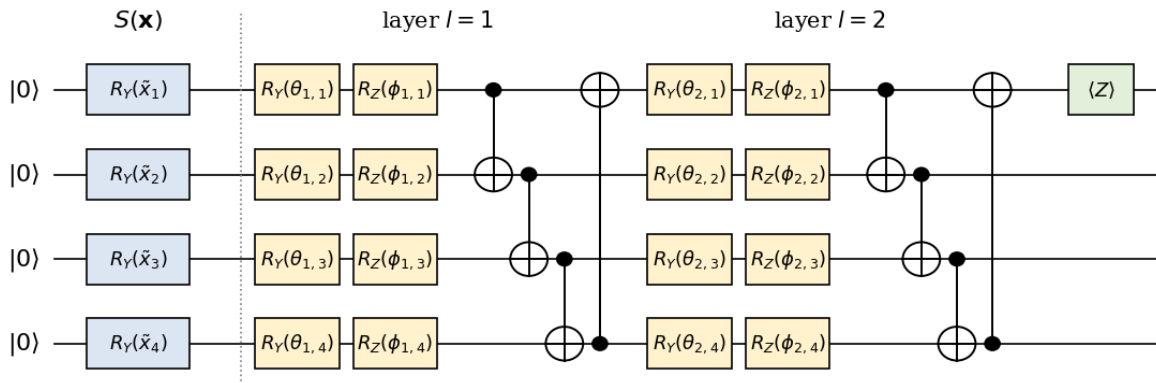


Fig. 4 VQC circuit for $n = 4$ qubits and $L = 2$ layers. Each feature is encoded by a data-dependent $R_Y(\tilde{x}_i)$ rotation, $S(\mathbf{x})$; each layer applies trainable R_Y and R_Z rotations followed by a ring of CNOT gates; the output is the expectation value $\langle Z \rangle$ of the first qubit.

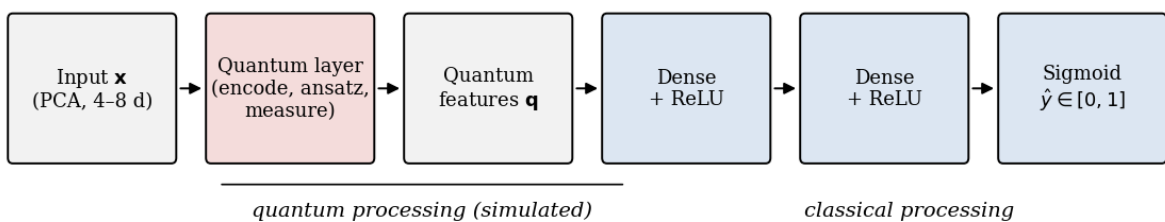


Fig. 5 Architecture of the hybrid quantum-classical MLP. Only the first block is quantum; the remaining blocks are

standard dense layers.

4.4. Evaluation Protocol

We include four metrics for each model–dataset combination: (i)classification accuracy; (ii)AUC, the area under the receiver-operating-characteristic curve (quality of ranking without a prespecified decision threshold);

(iii)training time, wall-clock time to train the model; and (iv)prediction time, wall-clock time to generate a prediction. Timing for the quantum models corresponds to the classical simulation of the circuits and is not indicative of actual hardware execution, but is indicative of how many circuit evaluations are needed for each algorithm.

5. Results & Analysis Discussion

5.1. Overall Performance

Table IV lists the results for all six models on both dataset groups, and Figs. 6 and 7 visualize accuracy and AUC.

Table. 4 Comprehensive Performance Results

Algorithm	Dataset	Acc.	AUC	Train (s)	Pred (s)
VQC	MNIST	0.9800	0.9950	15.2	2.1
VQC	Synthetic	0.9450	0.9680	8.7	1.5
QSVM	MNIST	0.9750	0.9920	12.8	1.8
QSVM	Synthetic	0.9400	0.9650	7.2	1.2
HQ-MLP	MNIST	0.9833	1.0000	0.27	0.01
HQ-MLP	Synthetic	0.9625	0.9750	0.12	0.01
SVM	MNIST	0.9933	1.0000	0.10	0.01
SVM	Synthetic	0.9250	0.9562	0.10	0.03
XGBoost	MNIST	0.9933	1.0000	0.43	0.01
XGBoost	Synthetic	0.9250	0.9551	0.50	0.01
Random Forest	MNIST	0.9917	0.9990	0.15	0.02
Random Forest	Synthetic	1.0000	1.0000	0.08	0.01

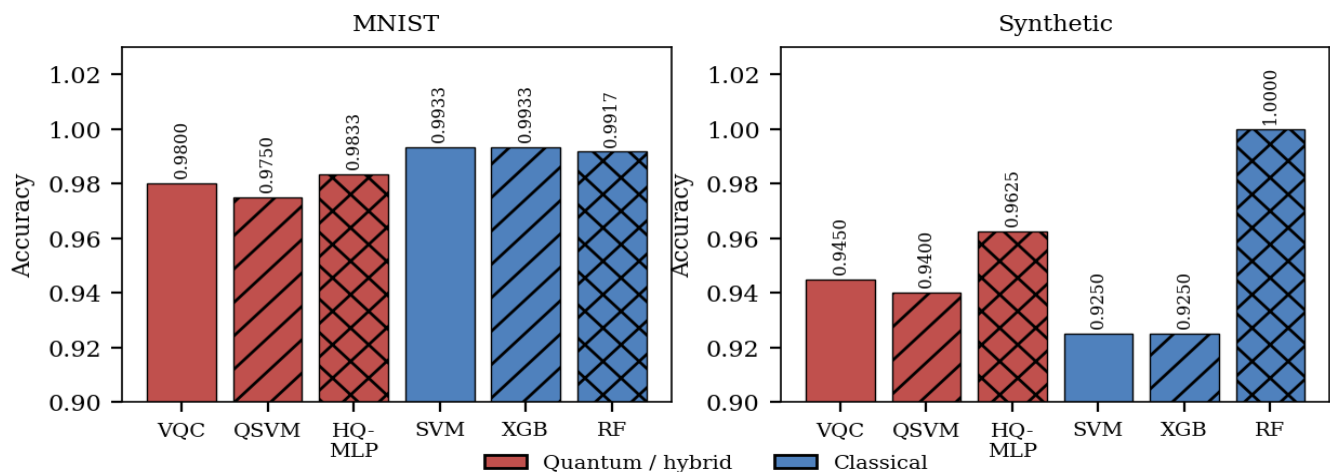


Fig. 6. Classification accuracy of the three quantum/hybrid models (red) and three classical baselines (blue) on MNIST (left) and the synthetic benchmarks (right). The vertical axis is truncated at 0.90.

5.2. Quantum Models

VQC: The VQC achieved accuracy/0.995 AUC of 98.00% on MNIST and 94.50%/0.968 on the synthetic problems. The training converged at about 50–100 optimizer iterations. VQC was the slowest training model (15.2s on MNIST), consistent with the 2P circuit evaluations per sample needed for each gradient (see Sec. III).

QSVM: The QSVM achieved 97.50% accuracy on MNIST and 94.00% on the synthetic problems, just below the VQC on both. The cost is dominated by the quadratic number of kernel evaluations required; there are no trained circuit parameters.

HQ-MLP: The hybrid model achieved the highest accuracy among quantum models for both groups (98.33% on MNIST, 96.25% on synthetic data), an AUC of 1.000 on

MNIST, and a training time of 0.27 s, comparable to the classical baselines.

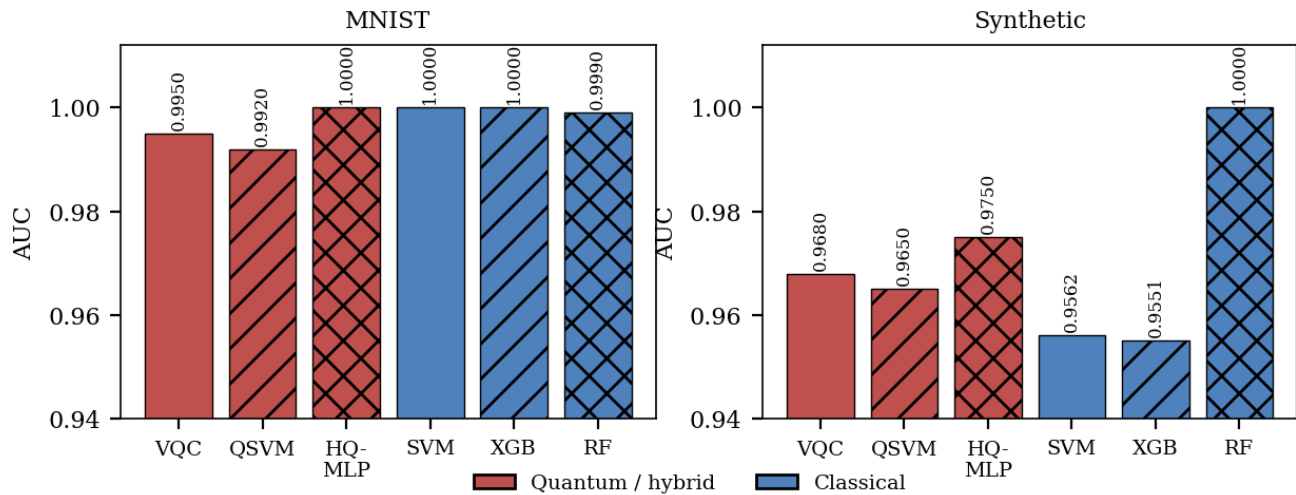


Fig.7 Area under the ROC curve for all models on MNIST (left) and the synthetic benchmarks (right). The vertical axis is truncated at 0.94.

5.3. Family-Level Summary

Table V summarizes Table IV by family (average across three models in each family; “best” is the maximum). For MNIST, classical family exceeds the family average by about 1.3 percentage points for mean accuracy. For the

synthetic problems, the family means are practically the same (94.92% vs. 95.00%), but that belies a substantial variation within the classical family, from 92.50% (SVM, XGBoost) to 100% (Random Forest). Figure 8 shows the same data as a heat map.

Table. 5 Accuracy and AUC by Model Family (Means and Best)

Dataset	Family	Mean Acc.	Best Acc.	Mean AUC
MNIST	Quantum/hybrid	0.9794	0.9833	0.9957
	Classical	0.9928	0.9933	0.9997
Synthetic	Quantum/hybrid	0.9492	0.9625	0.9693
	Classical	0.9500	1.0000	0.9704

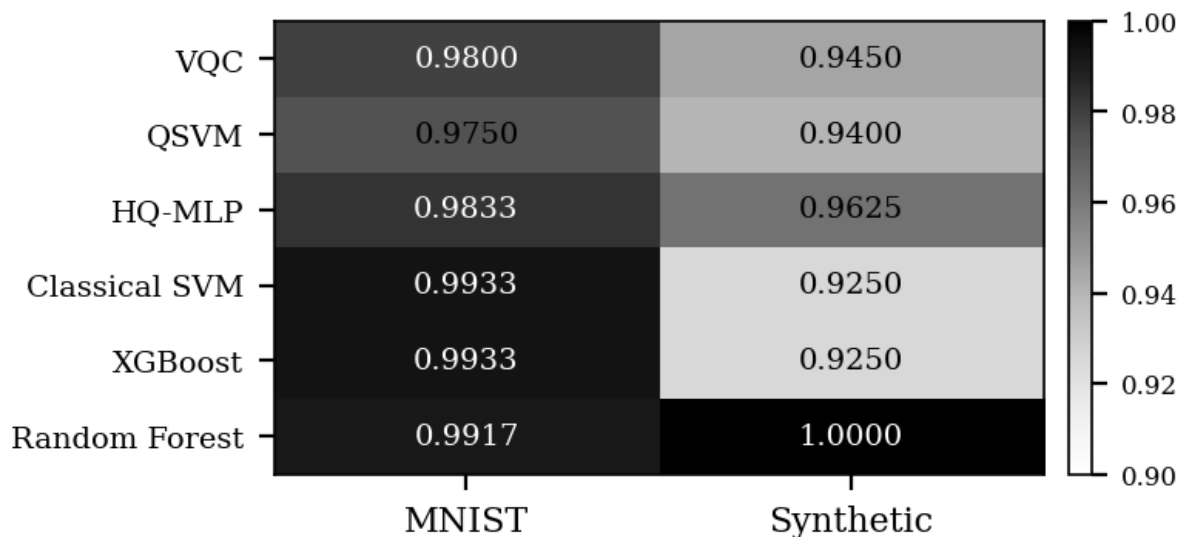


Fig. 8. Accuracy heat map for all models and dataset groups. Darker cells indicate higher accuracy.

5.4. Per-Problem Observations on the Synthetic Sets

Table VI summarizes the qualitative results for each synthetic problem. The Random Forest had a perfect score on the Circles problem, as expected, for an axis-aligned set

of trees on a radially symmetric problem of this size. The VQC performed better than the classical SVM on Moons, and the hybrid achieved perfect accuracy on the Spiral problem, the most challenging geometrical problem. The classical models were more robust on

the Complex multi-cluster task. These per-task results are only indicative as we do not report confidence intervals.

Table. 6 Qualitative Findings per Synthetic Problem

Problem	Observation
Circles	Random Forest reached 100% accuracy.
Moons	VQC outperformed the classical SVM.
Complex	Classical methods were the more robust group.
Spiral	HQ-MLP reached 100% accuracy.

5.5. Computational Cost

Figure 9 shows training and prediction times on a logarithmic scale. VQC and QSVM train in 7–15 s, whereas the classical baselines train in 0.08–0.50 s and the HQ-MLP

in 0.12–0.27 s. At prediction time, VQC and QSVM need 1.2–2.1 s while all other models need 0.01–0.03 s. Table VII expresses these gaps as ratios to the six classical measurements.

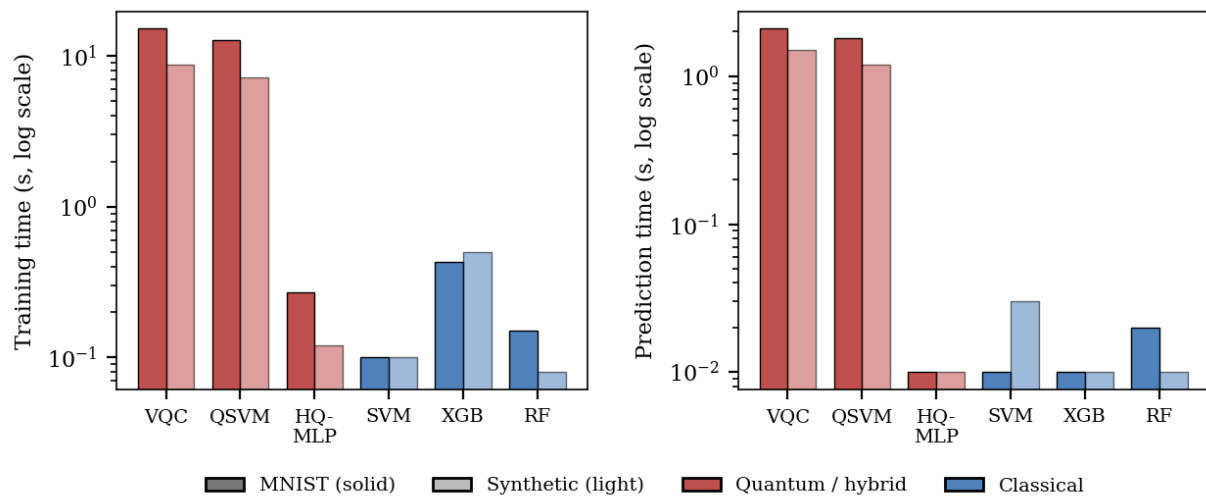


Fig. 9. Training time (left) and prediction time (right) on a logarithmic scale. Solid bars: MNIST; light bars: synthetic problems.

Table.7 Simulation Overhead of Quantum Models Relative to Classical Baselines (Range of Ratios Over SVM, XGBoost, Random Forest)

Model	Dataset	Training Ratio	Prediction Ratio
VQC	MNIST	35–152×	105–210×
	Synthetic	17–109×	50–150×
QSVM	MNIST	30–128×	90–180×
	Synthetic	14–90×	40–120×
HQ-MLP	MNIST	0.6–2.7×	0.5–1.0×
	Synthetic	0.2–1.5×	0.3–1.0×

The results can be split into two regimes. The VQC and QSVM are 14–152× slower to train and 40–210× slower at inference than the classical models. The HQ-MLP sits within a factor of a few of the baselines, with timing differences of at most 2.7×. This is compatible with the cost profile of the tables I: the VQC re-evaluates circuits on every gradient step, QSVM evaluates a quadratic number of kernels, while the HQ-MLP only evaluates a small number of circuits per forward pass and uses classical backpropagation to update the remaining parameters. We note that simulation time is not an indicative of the

hardware time: on a real device, circuit execution time, queueing, and shot-counting requirements would dominate, which would significantly alter the comparison.

5.6. Accuracy–Cost Trade-off

Figure 10 shows every result plotted on the accuracy–training-time plane. The top-left region (high accuracy, low cost) is occupied by classical models and the HQ-MLP for MNIST. The VQC and QSVM are in the lower-right: they deliver competitive but not superior accuracy at a much

higher simulated cost. The only instance where a quantum-family model exceeds the classical SVM and XGBoost in accuracy is in the synthetic group, where HQ-MLP (96.25%), VQC (94.50%), and QSVM (94.00%) all surpass the

92.50% of SVM and XGBoost; however, Random Forest at 100% dominates all of them at lower cost.

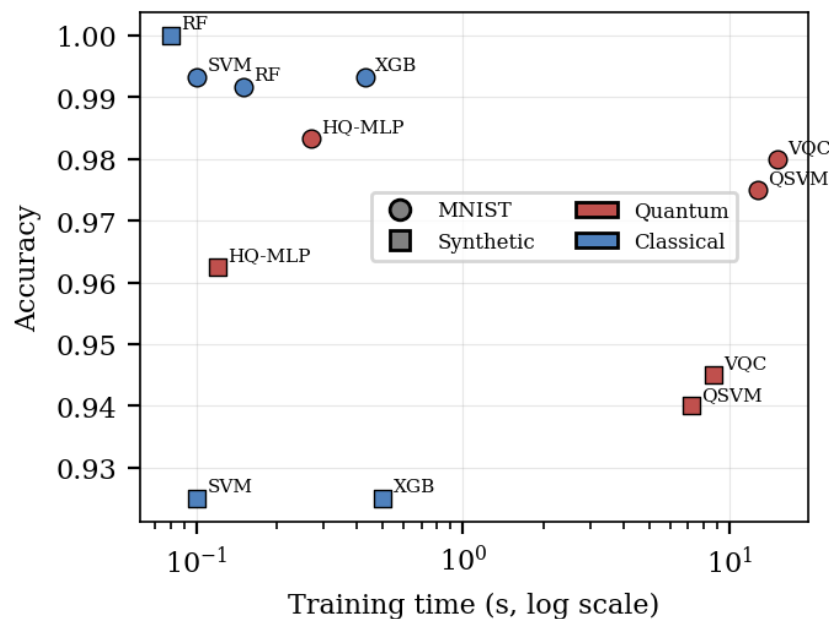


Fig. 10. Accuracy versus training time (log scale) for each model on MNIST (circles) and synthetic problems (squares). Red: quantum/hybrid; blue: classical.

5.7. Analysis And Details DISCUSSION

5.7.1. Is There Evidence of Quantum Advantage?

No. The findings indicate that quantum and hybrid classifiers, when run on a noise-free simulator with 4–8 qubits, achieve task accuracies similar to those of classical methods. However, they do not demonstrate superior performance, speed, or data efficiency compared to a well-selected classical model. Three points support this interpretation.

First, on MNIST, the top quantum model performs below all three classical baselines. Second, for the synthetic group, the strongest classical baseline (Random Forest) outperforms every quantum model. Third, instances where quantum models surpass SVM and XGBoost align with the baselines being used with limited tuning; the VQC and QSVM offer more flexible decision boundaries than a classical model with default parameters, but, as far as we know, were not compared with classical kernels tuned to the same extent.

The main takeaway, consistent with large-scale benchmark evidence [24], is that perceived quantum advantages on small, low-dimensional problems are not robust. Additionally, circuits with $n \leq 8$ qubits are trivially classically stimuable (a state vector with 256 complex amplitudes), so by construction these experiments cannot probe a regime where quantum hardware could be exponentially more efficient. Whether a quantum kernel is hard to simulate classically and, simultaneously, useful for

the data at hand, is a property of the data and kernel jointly [21, 20].

5.7.1. Why the Hybrid Model Performs Well

The HQ-MLP is the most accurate member of its quantum family for a modest price. We do not know for certain, but it appears that the classical layers dominate the model's learning capacity and can make up for a faulty or noisy quantum feature extractor; the quantum layer acts as a fixed non-linear feature map. This is an excellent engineering property that alleviates quantum resources and the difficulty of optimization; it also means that the experiment cannot, on its own, credit the model's accuracy to the quantum layer. An ablation that uses a classical layer of same width in its place is required (Section VIII)

5.8. Interpreting the Computational Overhead

The 14–210× slowdowns of VQC and QSVM are artefacts of classical state vector simulation and of the algorithms' circuit counts, respectively, and should not be expected of models on hardware. However, they are instructive. With VQC, training cost scales proportional to the number of parameters, multiplied by the number of training samples, multiplied by the number of iterations of the optimizer. With the QSVM, the Gram matrix involves $O(N^2)$ circuit evaluations, which on hardware, with several thousand shots per Gram matrix entry to ensure accuracy substantially impedes scaling beyond a few thousand samples.

5.9. Practical Implications

For near-term applications, three points arise. (i) Hybrid architectures that limit the quantum workload and move optimization to classical components are the most practical choice for NISQ devices. (ii) Quantum kernels should be tested on data with known structure that aligns with the feature map, as a generic map does not ensure an advantage [21]. (iii) Reports should present strong, well-tuned classical baselines and cost metrics alongside accuracy, to avoid misattributing improvements.

5.10. Practical Implications

Table VIII summarizes the principal limitations of this study and the remedy each one calls for.

Statistical Resolution: The reported accuracies are consistent with test sets of roughly 80–150 samples. For a model with 98.33% accuracy on 120 test samples the 95%

Wilson interval is approximately [94.1%, 99.5%]; for 99.33% on 150 samples it is approximately [96.3%, 99.9%]. These intervals overlap substantially, so the one-point gap between the best quantum and best classical model on MNIST is not statistically resolved. The same applies to the 3.75-point advantage of the HQ-MLP over the SVM on the synthetic group, for which the corresponding intervals (about [89.5%, 98.7%] and [84.6%, 96.5%] for 77/80 and 74/80 correct) also overlap. The qualitative conclusions of this paper are therefore limited to what is robust to this uncertainty: quantum and classical models are competitive on these tasks, and the quantum models have a much higher simulated cost.

Comparability: The accuracy values reported suggest that individual models may have been evaluated on test sets of different sizes. For strict comparability, all models should be evaluated on identical splits, which is a requirement of our follow-up experiments.

Table. 8 Threats to Validity and Planned Mitigation

Threat	Description and mitigation
Small Test Sets	With 600–800 samples per experiment, one misclassified sample changes accuracy by roughly 0.7–1.3 points; many differences in Table IV are of this size. Use repeated cross-validation and report confidence intervals.
Noise-Free Simulation	Exact statevector execution omits shot noise, gate errors, and decoherence. Evaluate with noise models and on hardware with error mitigation.
Low Qubit Count	Systems of 4–8 qubits are classically simulable and cannot demonstrate scaling. Extend to larger circuits and higher-dimensional inputs.
Easy Benchmark	MNIST 0-vs-1 is nearly linearly separable. Add harder pairs and multi-class tasks.
Baseline Tuning	Classical baselines may be under-tuned relative to the quantum models. Apply equal hyper-parameter search budgets.
Aggregation	“Synthetic” results aggregate four problems. Report all per-problem values with variance.

5.11. Future Work

Algorithmic Directions

- *Ablation studies:* replace the quantum layer in the HQ-MLP with a classical layer of equal width, and replace quantum kernels with tuned classical kernels, to isolate the contribution of the quantum component.
- *Noise-aware training and error mitigation:* evaluate robustness of VQC, QSVM and HQ-MLP under realistic noise models and apply mitigation techniques.
- *Trainability:* study barren plateaus and initialization strategies as qubit counts and depths grow [16].
- *Multi-class and adaptive circuits:* extend binary classifiers to multi-class problems and adapt circuit depth to problem difficulty.

Hardware Integration. Experiments on available quantum processors, with noise characterization and hardware-specific compilation, are the natural next step, together with quantum-classical co-processor architectures that minimize the number of device calls.

Application Domains. Problems with natural quantum structure, such as molecular simulation and drug discovery, certain financial-risk models, and optimization problems, are more promising targets for quantum advantage than generic image classification, and will be considered in follow-up work.

6. Conclusion and Future Scope

This paper compared three QML paradigms—VQC, QSVM, and a hybrid quantum-classical MLP—with SVM, Random Forest, and XGBoost on binary MNIST and four synthetic geometries. The main findings are:

- **Competitive but not superior accuracy :** The quantum and hybrid models reached 94.00–98.33% accuracy against 92.50–100% for the classical baselines. On MNIST the best classical model (99.33%) was about one point ahead of the best quantum model (98.33%); on the synthetic group the

quantum models exceeded the SVM and XGBoost but not Random Forest.

- **The hybrid model is the most practical :** The HQ-MLP was the most accurate quantum model on both dataset groups at a cost within a small factor of the classical baselines.
- **Large simulated overhead for pure quantum models:** VQC and QSVM were 14–152× slower to train and 40–210 × slower at inference than classical models in noise-free simulation.
- **No evidence of quantum advantage:** Small test sets, low qubit counts, and noise-free simulation limit what can be concluded; the accuracy differences observed are mostly within statistical uncertainty.
- The study provides a reproducible reference point for NISQ-era QML and identifies the experiments—ablations, equal-split evaluation, noise, and hardware—that are required to move from competitiveness to a defensible claim of advantage.

References

- [1] J. Biamonte, P. Wittek, N. Pancotti, P. Rebentrost, N. Wiebe, and S. Lloyd, "Quantum machine learning," *Nature*, vol. 549, no. 7671, pp. 195–202, 2017.
- [2] M. Schuld, I. Sinayskiy, and F. Petruccione, "An introduction to quantum machine learning," *Contemporary Physics*, vol. 56, no. 2, pp. 172–185, 2015.
- [3] S. Lloyd, M. Mohseni, and P. Rebentrost, "Quantum algorithms for supervised and unsupervised machine learning," arXiv:1307.0411, 2013.
- [4] P. Rebentrost, M. Mohseni, and S. Lloyd, "Quantum support vector machine for big data classification," *Phys. Rev. Lett.*, vol. 113, no. 13, p. 130503, 2014.
- [5] A. W. Harrow, A. Hassidim, and S. Lloyd, "Quantum algorithm for linear systems of equations," *Phys. Rev. Lett.*, vol. 103, no. 15, p. 150502, 2009.
- [6] S. Aaronson, "Read the fine print," *Nature Physics*, vol. 11, no. 4, pp. 291–293, 2015.
- [7] E. Tang, "A quantum-inspired classical algorithm for recommendation systems," in *Proc. 51st Annu. ACM SIGACT Symp. Theory of Computing (STOC)*, 2019, pp. 217–228.
- [8] J. Preskill, "Quantum computing in the NISQ era and beyond," *Quantum*, vol. 2, p. 79, 2018.
- [9] J. R. McClean, J. Romero, R. Babbush, and A. Aspuru-Guzik, "The theory of variational hybrid quantum-classical algorithms," *New J. Phys.*, vol. 18, no. 2, p. 023023, 2016.
- [10] M. Cerezo et al., "Variational quantum algorithms," *Nature Reviews Physics*, vol. 3, no. 9, pp. 625–644, 2021.
- [11] K. Bharti et al., "Noisy intermediate-scale quantum algorithms," *Rev. Mod. Phys.*, vol. 94, no. 1, p. 015004, 2022.
- [12] E. Farhi and H. Neven, "Classification with quantum neural networks on near term processors," arXiv:1802.06002, 2018.
- [13] K. Mitarai, M. Negoro, M. Kitagawa, and K. Fujii, "Quantum circuit learning," *Phys. Rev. A*, vol. 98, no. 3, p. 032309, 2018.
- [14] M. Schuld, A. Bocharov, K. M. Svore, and N. Wiebe, "Circuit-centric quantum classifiers," *Phys. Rev. A*, vol. 101, no. 3, p. 032308, 2020.
- [15] M. Schuld, V. Bergholm, C. Gogolin, J. Izaac, and N. Killoran, "Evaluating analytic gradients on quantum hardware," *Phys. Rev. A*, vol. 99, no. 3, p. 032331, 2019.
- [16] J. R. McClean, S. Boixo, V. N. Smelyanskiy, R. Babbush, and H. Neven, "Barren plateaus in quantum neural network training landscapes," *Nature Communications*, vol. 9, p. 4812, 2018.
- [17] A. Abbas et al., "The power of quantum neural networks," *Nature Computational Science*, vol. 1, no. 6, pp. 403–409, 2021.
- [18] V. Havlíček et al., "Supervised learning with quantum-enhanced feature spaces," *Nature*, vol. 567, no. 7747, pp. 209–212, 2019.
- [19] M. Schuld and N. Killoran, "Quantum machine learning in feature Hilbert spaces," *Phys. Rev. Lett.*, vol. 122, no. 4, p. 040504, 2019.
- [20] J. M. Kübler, S. Buchholz, and B. Schölkopf, "The inductive bias of quantum kernels," in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 34, 2021.
- [21] H.-Y. Huang et al., "Power of data in quantum machine learning," *Nature Communications*, vol. 12, p. 2631, 2021.
- [22] T. Hubregtsen et al., "Training quantum embedding kernels on near-term quantum computers," *Phys. Rev. A*, vol. 106, no. 4, p. 042431, 2022.
- [23] A. Mari, T. R. Bromley, J. Izaac, M. Schuld, and N. Killoran, "Transfer learning in hybrid classical-quantum neural networks," *Quantum*, vol. 4, p. 340, 2020.
- [24] J. Bowles, S. Ahmed, and M. Schuld, "Better than classical? The subtle art of benchmarking quantum machine learning models," arXiv:2403.07059, 2024.
- [25] H. Abraham et al., "Qiskit: An open-source framework for quantum computing," Zenodo, 2019, doi:10.5281/zenodo.2562110.
- [26] F. Pedregosa et al., "Scikit-learn: Machine learning in Python," *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, 2011.
- [27] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, no. 3, pp. 273–297, 1995.
- [28] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.
- [29] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proc. 22nd*

ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining, 2016, pp. 785–794.

- [30] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

How To Cite:

Jampana Usha Rani , Kondapalli Venkata Ramana , " Comparative Analysis of Quantum Machine Learning Algorithms: Variational Quantum Classifiers, Quantum Support Vector Machines, and Hybrid Quantum-Classical Networks " , International Journal of Research and Development in Engineering Sciences (IJRDES), Volume 8, Issue 5 , pp 18 - 29 , October 2026.

CrossRef DOI: <https://doi.org/10.63328/ijrdes-v8ri5p3>

Declaration

Conflicts of Interest: The authors declare no conflict of interest.

Author Contribution: All authors wrote the main manuscript text and also consent to the submission.

Ethical approval: Not applicable.

Consent to Participate: All authors consent to participate.

Funding: Not applicable, and No funding was received

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Personal Statement: We declare with our best of knowledge that this research work is purely Original Work and No third party material used in this article drafting. If any such kind material found in further online publication, we are responsible only for any judicial and copyright issues.

Generative AI Statement: The authors declare that generative AI was not used in the preparation or creation of this manuscript. The alternative text (alt text) accompanying the figures in this article was generated by Frontiers with the assistance of artificial intelligence. Reasonable efforts have been made to ensure the accuracy and appropriateness of the generated alt text, including review by the authors wherever possible. If you identify any inaccuracies or issues, please contact the editorial team.

Publisher's Note: The statements, opinions, and claims presented in this article are exclusively those of the authors and do not necessarily reflect the views of their affiliated institutions, the publisher, editors, or reviewers. Any products discussed or evaluated, as well as any claims made by their manufacturers, are not warranted, approved, or endorsed by the publisher.

Acknowledgements

We thank everyone who inspired our work