Indian Journal of Computer Science and Technology
https://www.doi.org/10.59256/indjcst.20260503131
Volume 5, Issue3 (September-December 2026), PP: 1094-1099.
www.indjcst.com
ISSN No: 2583-5300

Comparative Analysis of Classical Machine Learning and Quantum Machine Learning Algorithms

Rajat Singh[1], Vaibhav Kumar[2]
[1] *M.Tech Student, Department of Computer Science and Engineering, Bhagwant Institute of Technology, Muzaffarnagar, Uttar Pradesh, India.*
[2]*Assistant Professor, Department of Computer Science and Engineering, Bhagwant Institute of Technology, Muzaffarnagar, Uttar Pradesh, India.*

To Cite this Article: *Rajat Singh[1], Vaibhav Kumar[2], "Comparative Analysis of Classical Machine Learning and Quantum Machine Learning Algorithms", Indian Journal of Computer Science and Technology, Volume 05, Issue 03 (September-December 2026), PP: 1094-1099.*

Copyright: ©2026 This is an open access journal, and articles are distributed under the terms of the Creative Commons Attribution License; Which Permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

Abstract: *Quantum Machine Learning (QML) combines quantum circuits with classical learning procedures and has attracted interest as a possible approach to representing and processing data. However, current quantum devices and simulators operate under constraints that include limited qubit counts, circuit noise, finite measurement shots, data-encoding overhead, and substantial simulation cost. This paper presents a reproducible framework for comparing selected classical machine-learning (CML) and QML algorithms on compact tabular classification datasets. The classical baselines comprise Logistic Regression, Support Vector Machine (SVM), k-Nearest Neighbours (KNN), and Random Forest. The quantum methods comprise a Quantum Support Vector Classifier (QSVC) using a fidelity quantum kernel and a Variational Quantum Classifier (VQC) trained through a hybrid quantum-classical procedure. The planned evaluation uses public datasets including Iris and Breast Cancer Wisconsin (Diagnostic), with Wine as an optional dataset. All models will use common data partitions and controlled preprocessing. Predictive metrics will be considered alongside runtime, qubit count, circuit depth, gate count, measurement shots, and noise sensitivity. As implementation and experiments are not yet complete, this paper reports the research design and evaluation protocol, not measured performance or a claim of quantum advantage.*

Keywords: *Classical Machine Learning, Quantum Machine Learning, Quantum Computing, Quantum Support Vector Classifier, Variational Quantum Classifier, Quantum Kernel, Qiskit, Classification, Benchmarking.*

I. INTRODUCTION

Classical machine learning is widely used for classification, prediction, pattern recognition, and decision-support tasks. Algorithms such as Logistic Regression, Support Vector Machines, KNN, and Random Forest are supported by mature optimisation methods, well-tested software libraries, and conventional computing hardware [10]–[12]. Their performance on structured tabular datasets makes them essential baselines for evaluating any alternative learning paradigm.

Quantum computing uses quantum bits (qubits), quantum gates, superposition, entanglement, and measurement to represent and manipulate information. These principles motivate quantum algorithms for simulation, optimisation, and machine learning [1], [5]. In QML, classical features may be encoded into quantum states, processed by a circuit, and measured to obtain predictions or similarity values. Quantum kernel methods use a circuit to construct a feature-space representation, whereas variational quantum classifiers train parameterised circuits with the assistance of classical optimisers [2]–[4].

The practical value of QML remains an open empirical question. Near-term quantum devices are affected by imperfect gates, decoherence, restricted circuit depth, and finite measurement budgets. Simulator-based studies are useful for controlled experimentation, but simulator execution time is not equivalent to the runtime of a physical quantum processor [5]. Moreover, a high classification score on a small dataset does not establish a general quantum advantage. Recent benchmarking work highlights the importance of strong classical baselines, careful hyperparameter selection, and transparent accounting of data-loading and circuit costs [6], [7].

This study therefore proposes a fair comparison of representative CML and QML algorithms on small public tabular datasets. Its focus is not to demonstrate that quantum models are universally superior, but to identify how predictive performance, encoding choices, circuit complexity, noise, and computational cost vary across selected methods and datasets.

II. RELATED WORK AND RESEARCH CONTEXT

Foundational work by Biamonte et al. surveys the emerging relationship between quantum information and machine learning [1]. Schuld and Killoran connect quantum feature encoding with feature Hilbert spaces, providing a conceptual basis for quantum kernel methods [2]. Havlíček et al. demonstrate supervised learning with quantum-enhanced feature spaces and

show how quantum circuits can be used to construct non-classical feature representations [3]. These studies motivate the use of quantum feature maps in a controlled classification setting.

Variational quantum algorithms combine parameterised quantum circuits with classical optimisation. Cerezo et al. review their design and practical trainability challenges, including the importance of circuit structure and optimisation behaviour [4]. Such methods are relevant to VQC, where circuit parameters are updated to minimise a classification loss. The NISQ perspective also emphasises that current hardware constraints must shape algorithm design and interpretation [5].

For a fair comparison, classical methods must be tuned and evaluated with the same care as quantum methods. SVMs provide a particularly useful reference because QSVC also uses a kernel matrix and a classical support-vector optimisation procedure [10]. KNN and Random Forest represent distance-based and ensemble approaches, while Logistic Regression provides a simple linear baseline. Scikit-learn supplies a consistent implementation and evaluation environment for these estimators [12].

Recent empirical and review studies report mixed outcomes rather than a universal advantage for QML. Benchmarking analyses have shown that results can depend strongly on dataset properties, classical model selection, feature encoding, kernel configuration, and resource accounting [6]–[9]. The present work follows this evidence-based position and treats quantum performance as a question to be tested under predefined conditions.

III. RESEARCH GAP AND OBJECTIVES

Several methodological issues can weaken CML–QML comparisons: the use of only one classical baseline; unequal preprocessing or tuning budgets; single train/test splits; incomplete reporting of qubit, depth, shot, and runtime costs; and insufficient examination of encoding and noise sensitivity. A further concern is that quantum simulation costs may be misinterpreted as the runtime of quantum hardware. These issues motivate a compact but reproducible evaluation protocol.

The study addresses five research questions: RQ1—How do QSVC and VQC compare with Logistic Regression, SVM, KNN, and Random Forest on the same tabular datasets? RQ2—How do feature reduction and quantum data encoding affect predictive quality and circuit resources? RQ3—What trade-offs arise between predictive metrics and computational cost? RQ4—How sensitive are quantum models to circuit depth, measurement shots, and simulated noise? RQ5—Are observed differences consistent across datasets, random seeds, and cross-validation folds?

The objectives are to implement the selected classical and quantum classifiers; establish common preprocessing and data partitions; evaluate models with multiple classification metrics; measure computational and circuit resources; and analyse the stability of results through repeated experiments. The intended contribution is a student-manageable benchmark that reports both predictive outcomes and the additional requirements of quantum models.

IV. PROPOSED FRAMEWORK AND METHODOLOGY

The proposed workflow begins with public dataset selection, followed by data inspection, stratified splitting, training-only preprocessing, and feature reduction. The resulting compact feature vectors are supplied to classical models and to quantum circuits. Both model groups are evaluated on common folds, after which predictive metrics and computational-resource indicators are analysed together. The workflow is summarised in Figure 1.

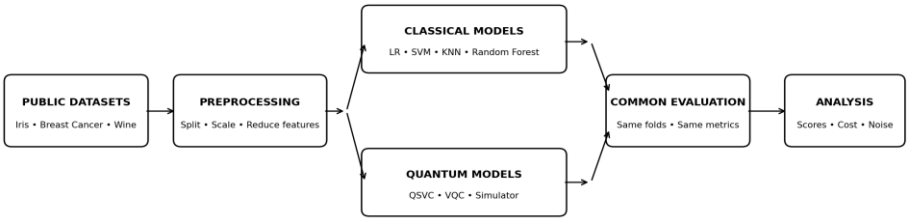


Figure 1. Proposed comparative framework for classical and quantum machine-learning evaluation.

4.1 Dataset selection and preprocessing

The primary candidate datasets are Iris and Breast Cancer Wisconsin (Diagnostic), with Wine included if execution time permits. These datasets are available through commonly used machine-learning repositories and libraries [15], [16]. They provide small-scale numerical classification tasks suitable for an initial simulator-based study. Their limited size is also a boundary: findings will not be generalised to large image, language, or industrial workloads.

The data pipeline will inspect class distributions, missing values, duplicate observations, feature types, and label definitions. Stratified train, validation, and test partitions or repeated stratified cross-validation will preserve class proportions. Scaling, feature selection, and Principal Component Analysis (PCA) will be fitted only on training data to prevent information leakage. Compact feature dimensions—such as two, four, and up to six features where feasible—will be considered to control the number of qubits and circuit evaluation cost.

Before modelling, the dataset inventory will record the source, version or retrieval date, number of observations, number of input features, target labels, and class frequencies. Duplicate rows and missing values will be checked and handled using rules defined before the final evaluation. The same label mapping and partition identifiers will be used across model families. Where class imbalance is present, stratified splitting and class-sensitive measures such as macro-F1 and balanced accuracy will be prioritised; any use of class weights will be documented. This audit is intended to make the experimental

inputs transparent and to reduce the possibility that differences arise from inconsistent data handling rather than the learning algorithms.

4.2 Classical machine-learning baselines

The classical comparison will include Logistic Regression, SVM with linear and radial-basis-function (RBF) kernels, KNN, and Random Forest. These methods represent linear, kernel-based, distance-based, and tree-ensemble learning. Hyperparameters will be selected using validation data or inner cross-validation. The final test data will remain untouched until model selection is complete. A documented and modest tuning budget will be applied to each baseline rather than relying only on default parameters.

Hyperparameter selection will be constrained to the training and validation data. For Logistic Regression, regularisation strength and solver settings will be recorded. For SVM, the penalty parameter and, for the RBF kernel, the kernel-width parameter will be considered. KNN experiments will record the neighbour count and distance metric, while Random Forest experiments will record the number of trees, maximum depth, and feature-sampling configuration. The final test partition will be used only after these choices have been fixed. This procedure is important because a weakly tuned classical model would provide an unfair reference for a more carefully configured QML model.

4.3 Quantum machine-learning models

QSVC will use a fidelity quantum kernel. A classical input x is encoded into a quantum state using a feature map $U\varphi(x)$, giving $|\varphi(x)\rangle = U\varphi(x)|0\rangle$. The similarity between two encoded inputs is estimated as $K(x_i, x_j) = |\langle\varphi(x_i)|\varphi(x_j)\rangle|^2$. The resulting kernel matrix is supplied to a support-vector classifier. The quantum circuit estimates similarities, while the SVM optimisation and decision function remain classical [2], [3].

$$\mathbf{K}(x_i, x_j) = |\varphi(x_i)|\varphi(x_j)|^2$$

VQC will use a feature-encoding circuit followed by a shallow trainable ansatz. A classical optimiser will update circuit parameters to minimise a classification loss. The number of qubits will be linked to the retained feature dimension. The initial implementation will use a small number of trainable layers; circuit depth and optimiser settings will be varied only within a documented experiment budget. Qiskit Machine Learning provides the intended implementation route for QSVC and VQC [13].

For quantum inputs, the primary encoding strategy will map scaled feature values to rotation gates such as RY and RZ operations. A shallow entangling feature map will be used for the fidelity kernel, while the VQC will combine an encoding circuit with a trainable ansatz. The retained feature count will determine the initial qubit requirement. Alternative encoding or feature-map settings will be tested only where simulator runtime allows. For QSVC, kernel construction requires pairwise comparisons between encoded samples; this can become a dominant cost as the sample count increases. The study will therefore record kernel construction time separately from classifier fitting and prediction. For VQC, training-loss behaviour and convergence will also be retained for error analysis.

Model family	Core approach	Role in the study
Classical ML	Logistic Regression, SVM, KNN, Random Forest	Establish strong conventional baselines.
QSVC	Fidelity quantum kernel with classical SVM optimisation	Evaluate quantum feature-space similarity.
VQC	Parameterized circuit trained with a classical optimiser	Evaluate hybrid variational learning.
Simulation	Ideal, finite-shot, and optional noisy simulation	Measure resource and noise sensitivity.

Table 1. Model families and their roles in the comparison.

4.4 Simulator configuration and reproducibility

A local quantum simulator will be the primary execution platform. An ideal simulator will establish a baseline; finite-shot simulation will be used to examine measurement effects, and a supported noise model may be introduced for a limited sensitivity analysis. Any optional cloud-hardware experiment will be reported separately. The paper will identify software versions, simulator backend, qubit count, circuit depth, gate count, shots, random seeds, processor environment, and configuration settings. Simulator wall-clock time will be treated as an implementation-dependent cost, not as intrinsic quantum runtime.

The implementation will proceed in stages. First, the classical pipeline will be validated using fixed data partitions and reproducible Scikit-learn configurations. Second, small quantum circuits will be tested on a limited sample to verify feature encoding, measurement, and prediction interfaces. Third, QSVC and VQC will be executed on the selected simulator using the same processed data. Only after these checks will repeated cross-validation and the planned sensitivity experiments be run. This staged approach reduces the risk that a software or circuit-construction error is mistaken for a model-performance difference.

4.5 Computational complexity and resource accounting

Resource accounting will distinguish the work performed by the classical host from the quantum-circuit workload. For QSVC, the number of pairwise kernel evaluations and the time required to construct training and test kernel matrices will

be recorded separately from SVM fitting. For VQC, the report will record the number of circuit evaluations required during training, the number of trainable parameters, measurement shots, and the stopping condition. Where supported by the simulator, circuit depth and gate counts will be collected before execution. These measurements will not be combined into a single unexplained runtime score; each will be reported with its unit and measurement context.

The same classical host and software environment will be used for model runs wherever practical. Timing will be repeated when variability is noticeable, and setup or compilation time will be distinguished from model training and prediction time when the platform exposes those stages. If a quantum kernel or circuit configuration exceeds the available execution budget, the limit will be documented and the affected comparison will be reported as incomplete. This avoids silently reducing the sample size or changing circuit settings in a way that would make the comparison difficult to reproduce.

V. EXPERIMENTAL DESIGN AND EVALUATION

The experimental plan separates six comparisons: (E1) classical baseline performance; (E2) QSVC against linear and RBF SVM; (E3) VQC against the classical models; (E4) sensitivity to retained feature dimensions and encoding; (E5) sensitivity to circuit depth and shot count; and (E6) ideal versus finite-shot or noisy simulation. Each experiment will have a predefined question, dataset partition, configuration log, and metric set.

5.1 Evaluation metrics

Predictive performance will be assessed using accuracy, precision, recall, macro-F1, balanced accuracy, and confusion matrices. Macro-F1 will receive particular attention for multiclass tasks because it gives equal weight to each class. ROC-AUC will be reported for binary classification when suitable probability or decision scores are available. Accuracy will not be used as the sole measure, especially when class distributions are unequal.

Computational cost will include training time, prediction time, and—where applicable—quantum-kernel construction time. Quantum resource reporting will include qubit count, circuit depth, gate count, measurement shots, and simulator settings. Memory use may also be recorded. These measures will help distinguish a model with competitive predictive metrics from one that requires substantially greater execution or circuit resources.

5.2 Statistical analysis

Models will be compared using paired results from the same folds or repeated random seeds. Descriptive summaries will include mean, standard deviation, and confidence intervals where feasible. For two-model comparisons, a Wilcoxon signed-rank test may be considered when parametric assumptions are not justified. For comparisons involving more than two models, a Friedman test with an appropriate post-hoc procedure may be used. Multiple-comparison correction will be applied when several pairwise tests are conducted.

Statistical significance will not be interpreted as practical importance. Absolute metric differences and effect sizes will be reported alongside p-values. If the number of folds, seeds, or datasets is too small for robust inference, the analysis will be described as exploratory. Failed or interrupted runs will be logged rather than silently excluded.

5.3 Reproducibility and Validity Controls

Reproducibility will be supported by a version-controlled project structure and a configuration file defining the dataset, random seed, fold identifiers, preprocessing steps, model parameters, quantum feature map, ansatz, simulator backend, shot count, and noise setting. Each run will store predictions, true labels, fold and run identifiers, timing information, and circuit metadata. Software versions and the host processor and memory configuration will also be recorded. A failed run, timeout, or convergence problem will be documented with its configuration so that unsuccessful experiments remain visible in the final account.

The main threats to validity are dataset representativeness, feature-reduction effects, hyperparameter sensitivity, simulator dependence, and statistical uncertainty. The selected benchmarks are small and may favour some model families. PCA can reduce the qubit requirement but may discard information relevant to a class boundary. The selected feature map and ansatz represent only a small portion of the QML design space. These limitations will be addressed by reporting dataset-level outcomes separately, repeating selected runs with fixed seeds, and describing conclusions as specific to the evaluated configurations rather than universal properties of QML.

5.4 Error analysis and ablation plan

Error analysis will examine confusion matrices and class-wise precision and recall to identify whether a model performs unevenly across classes. For the classical models, errors will be considered in relation to feature scaling, class overlap, model capacity, and selected hyperparameters. For QSVC, the study will inspect whether the estimated kernel matrix contains nearly identical similarities or numerical behaviour that may limit discrimination. For VQC, training-loss curves, convergence behaviour, and variation across initial parameter seeds will be reviewed. These checks will help distinguish a poor model result from an unstable or incomplete implementation.

Ablation experiments will change one factor at a time while keeping the dataset partition fixed. The planned factors include retained feature dimension, quantum feature-map configuration, VQC ansatz depth, measurement-shot count, and ideal versus noisy simulation. The feature-dimension comparison will consider compact inputs such as two, four, and up to six features where feasible. Circuit-depth tests will compare a shallow configuration with a moderately deeper alternative, rather than searching an unrestricted circuit space. This controlled approach will make it easier to attribute observed changes to a specific design choice.

5.5 Results organisation and reporting protocol

Results will be organised by dataset before any cross-dataset summary is presented. Each dataset-level report will identify the number of observations, feature dimensions used, class distribution, split strategy, and final model configuration. A comparison table will report the predictive metrics for every completed model, while a separate resource table will report runtime and quantum-circuit indicators. This separation is important because a model may have a competitive macro-F1 score while requiring a substantially larger number of circuit evaluations or a longer kernel-construction stage.

The report will include the mean and standard deviation across repeated folds or seeds, together with confidence intervals where the available sample supports them. Incomplete runs, unavailable measurements, and optional hardware results will be labelled explicitly. The final interpretation will distinguish descriptive differences, statistically supported differences, and practical significance. No single aggregate ranking will conceal dataset-specific variation, and no result will be generalised beyond the tested algorithms, datasets, and configurations.

VI. DISCUSSION

The framework treats CML and QML as alternative experimental approaches to the same supervised-learning task. Its main methodological principle is fairness: models should receive equivalent data partitions, label definitions, preprocessing rules, and transparent tuning budgets. The use of both QSVC and VQC provides coverage of two distinct QML families, while the classical baselines prevent the comparison from depending on a single conventional algorithm.

The expected value of the study lies in explaining the conditions behind observed outcomes. A QML model may be competitive on one dataset but not another; a particular encoding may improve a metric while increasing circuit depth; or noise and finite shots may reduce stability. Conversely, classical methods may perform as well as or better than the quantum models with substantially lower computational overhead. Each of these outcomes is informative when the protocol is reproducible. No improvement, speedup, or quantum advantage is claimed before the planned experiments are completed.

Interpretation will consider both predictive quality and the resources required to obtain it. If QSVC or VQC achieves a similar score to a classical model but requires more simulation time, circuit evaluations, or measurement shots, the result will be described as predictive competitiveness without a demonstrated efficiency benefit. If a quantum model performs better on one dataset, the discussion will examine whether the result persists across folds and seeds and whether it is sensitive to feature-map settings. If classical models perform better, that outcome will be reported directly rather than attributed to an assumed implementation defect. The study is designed to produce a balanced comparison, including parity and negative results.

An important consideration is the cost of representing classical data in a quantum state. Data encoding is not a free operation: the chosen feature map, number of retained dimensions, circuit depth, and measurement strategy all influence the workload. Dimensionality reduction may make a quantum experiment feasible, but it also changes the information supplied to every model. For this reason, the primary comparison will use matched processed features, and any secondary comparison using a different feature representation will be identified as a separate experiment rather than mixed with the main result.

The study will also distinguish the two quantum model families. QSVC relies on a quantum kernel and a classical SVM decision procedure; its practical cost may be dominated by repeated state-overlap evaluations. VQC instead learns circuit parameters through repeated forward evaluations and classical optimisation; its behaviour may depend on ansatz depth, initialisation, optimiser settings, and noise. A single statement that 'QML works' or 'QML does not work' would therefore be too broad. The useful conclusion will describe which method was evaluated, under what data and resource conditions, and how stable its outcome was.

VII. LIMITATIONS AND ETHICAL CONSIDERATIONS

The selected datasets are small public benchmarks and may not represent complex real-world tasks. Feature reduction can remove useful information, and results may depend on the chosen feature map, ansatz, optimiser, simulator, and tuning budget. Quantum simulation time depends on the classical host and software implementation; it cannot be equated with the runtime of a physical quantum processor. Noise models are approximations and may not capture all device behaviour. These limitations will be stated alongside the results.

The study will use public datasets under their applicable terms and attribution requirements. The Breast Cancer Wisconsin dataset will be treated solely as a benchmark for algorithm comparison, not as a clinically validated diagnostic system. The work will not make decisions about individuals. Responsible reporting will include negative and inconclusive findings and will avoid claims of general quantum supremacy or universal advantage.

A further interpretive distinction concerns algorithmic and hardware claims. A simulator can help assess the mathematical behaviour of a circuit under controlled assumptions, but it does not reproduce every limitation of a physical device. Likewise, a small optional cloud execution may demonstrate feasibility but cannot establish a general hardware advantage. The final paper will therefore separate classification quality, simulator cost, circuit-resource requirements, and any optional hardware observations in both tables and discussion.

The statistical scope of the study is also limited by the number and size of the available benchmark datasets. Repeated folds provide more information than a single split, but they do not create independent real-world datasets. If the number of datasets or repeated runs is insufficient for a strong inferential claim, statistical tests will be treated as exploratory. Differences in package versions, simulator implementations, and hardware environments may also affect reproducibility; these factors will be documented so that future replications can identify whether a discrepancy arises from the method or the execution environment.

VIII. CONCLUSION AND FUTURE WORK

This paper presents a reproducible research framework for comparing selected classical machine-learning algorithms with QSVC and VQC on small public tabular classification datasets. The proposed design combines common preprocessing and evaluation partitions with multiple predictive metrics, resource accounting, circuit and encoding sensitivity tests, and repeated statistical analysis. Its purpose is to provide an evidence-based assessment of when selected QML approaches are competitive and what additional constraints they introduce.

The next stage is implementation of the dataset pipeline, classical baselines, and quantum classifiers, followed by the planned experiments. Future extensions may examine larger datasets, additional encodings, improved noise models, hardware execution, and domain-specific tasks. Any conclusions will remain limited to the tested algorithms, datasets, and resource settings.

The practical deliverables will include a documented dataset pipeline, executable classical baselines, QSVC and VQC simulator implementations, experiment logs, metric summaries, resource reports, and a reproducible configuration record. The evidence will support a bounded conclusion about the selected datasets and algorithms. Larger-scale experiments, additional quantum encodings, hardware execution, and domain-specific applications are appropriate future extensions, but they are outside the required scope of this initial M.Tech study.

The intended contribution is methodological rather than a claim of a new quantum algorithm. It is the combination of strong classical baselines, two representative QML families, controlled data preparation, repeated evaluation, explicit circuit-resource reporting, and cautious interpretation within one reproducible M.Tech-scale study. Such a design can provide a useful starting point for later work that expands the dataset range, tests additional feature maps, or evaluates selected circuits on physical quantum devices.

REFERENCES

1. J. Biamonte, P. Wittek, N. Pancotti, P. Rebentrost, N. Wiebe, and S. Lloyd, “Quantum machine learning,” Nature, vol. 549, pp. 195–202, 2017. doi: 10.1038/nature23474.
2. M. Schuld and N. Killoran, “Quantum machine learning in feature Hilbert spaces,” Physical Review Letters, vol. 122, art. 040504, 2019. doi: 10.1103/PhysRevLett.122.040504.
3. V. Havlíček et al., “Supervised learning with quantum-enhanced feature spaces,” Nature, vol. 567, pp. 209–212, 2019. doi: 10.1038/s41586-019-0980-2.
4. M. Cerezo et al., “Variational quantum algorithms,” Nature Reviews Physics, vol. 3, pp. 625–644, 2021. doi: 10.1038/s42254-021-00348-9.
5. J. Preskill, “Quantum computing in the NISQ era and beyond,” Quantum, vol. 2, art. 79, 2018. doi: 10.22331/q-2018-08-06-79.
6. J. Bowles, S. Ahmed, and M. Schuld, “Better than classical? The subtle art of benchmarking quantum machine learning models,” arXiv:2403.07059, 2024.
7. J. Schnabel and M. Roth, “Quantum kernel methods under scrutiny: a benchmarking study,” Quantum Machine Intelligence, vol. 7, art. 58, 2025. doi: 10.1007/s42484-025-00273-5.
8. S. S. Ajibosin and D. Cetinkaya, “Implementation and performance evaluation of quantum machine learning algorithms for binary classification,” Software, vol. 3, no. 4, pp. 498–513, 2024.
9. D. Ranga et al., “Quantum machine learning: exploring the role of data encoding techniques, challenges, and future directions,” Mathematics, vol. 12, no. 21, art. 3318, 2024.
10. C. Cortes and V. Vapnik, “Support-vector networks,” Machine Learning, vol. 20, pp. 273–297, 1995. doi: 10.1007/BF00994018.
11. L. Breiman, “Random forests,” Machine Learning, vol. 45, pp. 5–32, 2001. doi: 10.1023/A:1010933404324.
12. F. Pedregosa et al., “Scikit-learn: machine learning in Python,” Journal of Machine Learning Research, vol. 12, pp. 2825–2830, 2011.
13. Qiskit Community, “Qiskit Machine Learning: QSVC and VQC API documentation,” accessed November 2026. Available: https://qiskit-community.github.io/qiskit-machine-learning/.
14. PennyLane Developers, “Variational classifier,” PennyLane demonstrations, last updated April 17, 2026. Available: https://pennylane.ai/demos/tutorial_variational_classifier.
15. Scikit-learn Developers, “Toy datasets and dataset loading utilities,” Scikit-learn documentation, accessed November 2026. Available: https://scikit-learn.org/stable/datasets/toy_dataset.html.
16. University of California, Irvine, “UCI Machine Learning Repository,” accessed November 2026. Available: https://archive.ics.uci.edu/.