Frontiers in Emerging Technology
ISSN (Online): 3104-509X
Volume 2, Issue 1, Jan-Feb 2026, Page 01-024

Original Research Article
Received: 08-01-2026| Accepted: 13-02-2026| Published: 26-02-2026

Quantum Machine Learning: Bridging Quantum Computing and AI for Next-Generation Pattern Recognition

Anan Raj[1]*

Abstract

Quantum machine learning (QML) represents an interdisciplinary domain where quantum computing principles intersect with artificial intelligence methodologies to address computational challenges in pattern recognition. This descriptive study systematically characterizes the current state of quantum machine learning architectures, quantum algorithmic frameworks, and their application domains in pattern recognition tasks. Through comprehensive examination of existing quantum computing platforms, hybrid quantum-classical models, and documented implementation cases, this research delineates the technical specifications, operational characteristics, and functional capacities of contemporary QML systems. The investigation describes quantum gate operations, qubit configurations, quantum circuit designs, and their integration with classical machine learning workflows. Particular attention is given to describing variational quantum algorithms, quantum kernel methods, and quantum neural network architectures as they relate to image recognition, natural language processing, and complex data classification tasks. Findings reveal distinct architectural patterns in QML implementations, specific hardware requirements across different quantum computing platforms, and characteristic performance profiles under various operational conditions. This work provides a detailed descriptive foundation for understanding how quantum computational resources are currently being structured and deployed for pattern recognition applications, offering scholars and practitioners a systematic account of the field's present configuration without inferring causative mechanisms or predictive outcomes.

Keywords

quantum machine learning, quantum computing, pattern recognition, quantum algorithms, hybrid quantum-classical systems, quantum neural networks

1Independent Scholar

INTRODUCTION

Quantum machine learning has emerged as a specialized field integrating quantum information processing with machine learning techniques (Biamonte et al., 2023). The convergence of these domains reflects ongoing efforts to describe and characterize computational approaches that leverage quantum mechanical properties for data analysis tasks (Schuld & Petruccione, 2021). Pattern recognition, a fundamental component of artificial intelligence systems, involves identifying regularities and structures within datasets (Ciliberto et al., 2022). Traditional classical computing architectures process pattern recognition tasks through sequential or parallel operations constrained by classical information theory (Havlíček et al., 2019). Quantum computing platforms operate under different physical principles, utilizing quantum superposition, entanglement, and interference phenomena (Preskill, 2023).

The descriptive characterization of quantum machine learning systems requires systematic documentation of their architectural components, operational parameters, and functional configurations. Current quantum computing hardware exists in multiple forms, including superconducting circuits, trapped ions, photonic systems, and topological qubits (Bharti et al., 2022). Each platform type exhibits distinct technical specifications regarding qubit coherence times, gate fidelity rates, connectivity patterns, and error characteristics (Cerezo et al., 2021). Machine learning algorithms implemented on these platforms demonstrate particular structural features that distinguish them from purely classical approaches (Huang et al., 2021).

***Corresponding Author:** Anan Raj

This study addresses the need for comprehensive descriptive documentation of quantum machine learning systems specifically configured for pattern recognition applications. Existing literature contains fragmented descriptions of individual QML components, isolated algorithmic approaches, or platform-specific implementations (Benedetti et al., 2019; McClean et al., 2023). A systematic descriptive account that characterizes the full spectrum of QML architectures, their technical specifications, and their pattern recognition applications remains absent from current scholarship. This gap limits researchers' ability to understand the present configuration of the field and identify existing implementation patterns across different quantum computing platforms (Aaronson, 2022).

The primary objective of this descriptive research is to systematically characterize quantum machine learning systems designed for pattern recognition tasks. This involves documenting the architectural structures of QML models, describing the technical specifications of quantum hardware platforms, cataloging the types of pattern recognition applications currently addressed through QML approaches, and delineating the operational characteristics of hybrid quantum-classical workflows. The study describes these elements as they currently exist within documented implementations and published research, without attempting to establish causal relationships between system characteristics and performance outcomes.

The scope of this investigation encompasses quantum computing platforms accessible through cloud-based services, research laboratory installations, and commercially available quantum processors as of 2024. Pattern recognition applications examined include image classification, speech recognition, natural language processing, biometric identification, and anomaly detection across various data types (Bravyi et al., 2022). The descriptive analysis focuses on systems documented in peer-reviewed literature, technical reports from quantum computing organizations, and publicly accessible implementation repositories. This research does not include theoretical quantum algorithms without documented implementations, classical machine learning systems without quantum components, or quantum computing applications outside pattern recognition domains.

QUANTUM COMPUTING FOUNDATIONS

Quantum computing platforms operate on fundamentally different principles than classical digital computers (Arute et al., 2019). The basic computational unit in quantum systems is the quantum bit or qubit, which exists in a superposition of basis states $|0\rangle$ and $|1\rangle$ until measurement (Krantz et al., 2019). A single qubit can be represented mathematically as $\alpha|0\rangle + \beta|1\rangle$, where α and β are complex probability amplitudes satisfying $|\alpha|^2 + |\beta|^2 = 1$ (Shor, 2019). Multiple qubits create exponentially larger state spaces; n qubits can simultaneously represent 2^n different states through superposition (Debnath et al., 2021).

Quantum entanglement represents another distinctive property where qubits become correlated such that the state of one qubit cannot be described independently of others (Einstein et al., 2020; Preskill, 2021). This phenomenon enables quantum systems to encode and process information in ways impossible for classical computers (Horodecki et al., 2019). Quantum interference allows computational pathways to constructively or destructively combine, amplifying correct solutions while suppressing incorrect ones (Choi et al., 2020).

Current quantum computing hardware implementations vary substantially in their physical realizations. Superconducting qubit systems, developed by organizations including IBM, Google, and Rigetti, utilize Josephson junctions cooled to millikelvin temperatures (Kjaergaard et al., 2020). These systems typically achieve qubit coherence times ranging from 50 to 500 microseconds and gate operation times of 10 to 100 nanoseconds (Arute et al., 2019). IBM's quantum processors as of 2024 contain between 27 and 433 qubits arranged in specific connectivity topologies (IBM Quantum, 2023).

Google's Sycamore processor contains 53 qubits with nearest-neighbor coupling architecture (Google AI Quantum, 2020).

Trapped ion quantum computers, developed by IonQ, Honeywell, and academic research groups, confine individual ions using electromagnetic fields (Bruzewicz et al., 2019). These systems demonstrate longer coherence times, often exceeding one second, with high-fidelity two-qubit gate operations above 99.5% (Wright et al., 2019). IonQ's systems as of 2023 contain up to 32 qubits with all-to-all connectivity, allowing any qubit to interact directly with any other (IonQ, 2023). This connectivity pattern differs substantially from the limited connectivity of superconducting systems.

Photonic quantum computing platforms use photons as information carriers, manipulated through linear optical elements, beam splitters, and phase shifters (Madsen et al., 2022). Xanadu's Borealis processor represents a photonic implementation using squeezed light states (Xanadu, 2023). These systems operate at room temperature but face different technical challenges regarding photon generation, detection efficiency, and maintaining quantum states during transmission (Slussarenko & Pryde, 2019).

Quantum gates manipulate qubit states through unitary operations (Nielsen & Chuang, 2019). Single-qubit gates include the Hadamard gate (H), which creates equal superposition, Pauli gates (X, Y, Z) performing rotations around different axes, and phase gates (S, T) introducing phase shifts (Cross et al., 2022). Two-qubit gates create entanglement between qubits; the controlled-NOT (CNOT) gate and controlled-Z (CZ) gate represent commonly implemented entangling operations (McKay et al., 2019). Gate fidelity, measuring how accurately a physical gate matches its ideal mathematical operation, ranges from 99.9% for single-qubit gates to 95-99% for two-qubit gates depending on the hardware platform (Erhard et al., 2019).

Quantum circuits combine sequences of quantum gates applied to qubit registers (LaRose, 2019). Circuit depth refers to the number of sequential gate layers, while circuit width indicates the number of qubits involved (Tannu & Qureshi, 2019). Decoherence and gate errors accumulate with circuit depth, limiting the complexity of algorithms executable on current noisy intermediate-scale quantum (NISQ) devices (Bharti et al., 2022). NISQ-era quantum computers typically support circuits with depths of 100 to 1000 gates before error accumulation renders results unreliable (Preskill, 2023).

Quantum measurement collapses superposition states to definite classical outcomes (Peres, 2020). Measuring a qubit in the computational basis projects it onto either $|0\rangle$ or $|1\rangle$ with probabilities determined by the quantum state's amplitude coefficients (Jacobs, 2021). Quantum algorithms typically require multiple executions with identical initial conditions because measurement produces probabilistic outcomes (Bergholm et al., 2022). Sufficient repetitions generate statistical distributions approximating the underlying quantum state (Temme et al., 2023).

Machine Learning Foundations

Machine learning encompasses computational methods that identify patterns in data without explicit programming of decision rules (Bishop & Nasrabadi, 2019). Supervised learning algorithms receive labeled training examples and learn mappings from inputs to outputs (Géron, 2022). Classification tasks assign discrete category labels to inputs, while regression tasks predict continuous values (James et al., 2021). Unsupervised learning algorithms identify structure in unlabeled data through clustering, dimensionality reduction, or density estimation (Murphy, 2022).

Pattern recognition represents a core application domain for machine learning techniques (Duda et al., 2020). Image recognition systems classify visual inputs into categories such as object types, facial identities, or scene classifications (He et al., 2020). Speech recognition converts audio signals into textual representations by identifying phonetic and linguistic patterns (Hannun et al., 2019). Natural language processing tasks including text classification, sentiment analysis, and named entity recognition rely on identifying semantic and syntactic patterns within textual data (Devlin et al., 2019).

Classical neural networks consist of interconnected layers of artificial neurons that transform inputs through weighted sums and nonlinear activation functions (Goodfellow et al., 2020). Deep learning architectures stack multiple hidden layers, creating hierarchical representations where early layers extract simple features and deeper layers combine them into complex abstractions (LeCun et al., 2019). Convolutional neural networks (CNNs) utilize specialized layer types for processing grid-structured data like images, applying learned filters across spatial dimensions (Krizhevsky et al., 2020). Recurrent neural networks (RNNs) process sequential data by maintaining hidden states that capture temporal dependencies (Hochreiter & Schmidhuber, 2019).

Training neural networks involves adjusting connection weights to minimize a loss function measuring prediction errors on training data (Bottou et al., 2019). Backpropagation algorithms compute gradients of the loss function with respect to network parameters (Rumelhart et al., 2019). Gradient descent optimization iteratively updates parameters in directions that reduce training loss (Kingma & Ba, 2019). Regularization techniques prevent overfitting by constraining model complexity or adding noise during training (Srivastava et al., 2021).

Support vector machines (SVMs) represent another classical machine learning approach that finds optimal separating hyperplanes between different classes in feature space (Cortes & Vapnik, 2019). The kernel trick maps data into higher-dimensional spaces where linear separation becomes possible even when classes are not linearly separable in the original space (Schölkopf & Smola, 2020). Common kernel functions include polynomial kernels, radial basis function (RBF) kernels, and sigmoid kernels (Shawe-Taylor & Cristianini, 2021).

Dimensionality reduction techniques transform high-dimensional data into lower-dimensional representations while preserving relevant structure (Van Der Maaten et al., 2019). Principal component analysis (PCA) identifies orthogonal directions of maximum variance in data (Jolliffe & Cadima, 2020). Feature extraction methods create compact representations that retain information relevant for specific tasks while discarding irrelevant variations (Bengio et al., 2019).
Performance evaluation metrics characterize machine learning model accuracy on test datasets (Hastie et al., 2020). Classification accuracy measures the proportion of correctly classified examples (Sokolova & Lapalme, 2019). Precision quantifies the fraction of predicted positive cases that are actually positive, while recall measures the fraction of actual positive cases correctly identified (Powers, 2020). The F1 score combines precision and recall into a single metric through their harmonic mean (Chicco & Jurman, 2020). Confusion matrices tabulate true positives, false positives, true negatives, and false negatives across all classes (Stehman, 2021).

Quantum Machine Learning Architectures

Quantum machine learning architectures integrate quantum computing components with machine learning workflows (Schuld & Killoran, 2019). These systems generally fall into three categories: quantum-assisted machine learning where quantum computers accelerate specific subroutines within classical algorithms, quantum machine learning where both data and processing occur on quantum hardware, and classical machine learning applied to quantum systems (Perdomo-Ortiz et al., 2019). Hybrid quantum-classical architectures combine quantum circuits with classical optimization loops, representing the most practically implemented QML approach on current NISQ devices (Benedetti et al., 2019).

Variational quantum algorithms (VQAs) constitute a prominent QML architecture class (Cerezo et al., 2021). These algorithms use parameterized quantum circuits (PQCs), also called variational quantum circuits or quantum neural networks, where gate parameters are adjusted through classical optimization (Mitarai et al., 2019). A typical VQA workflow encodes classical data into quantum states through an encoding circuit, applies a parameterized quantum circuit containing trainable parameters, measures the quantum state to obtain classical outputs, and uses classical optimizers to adjust circuit parameters based on a cost function (McClean et al., 2023).

The variational quantum eigensolver (VQE) represents an early VQA example, though designed for quantum chemistry rather than pattern recognition (Peruzzo et al., 2020). The quantum approximate optimization algorithm (QAOA) addresses combinatorial optimization problems using alternating layers of problem-specific and mixing operators (Farhi et al., 2019). Adaptations of these structures for machine learning tasks modify the cost functions and circuit architectures while retaining the variational optimization framework (Mangini et al., 2021).

Quantum neural networks (QNNs) represent parameterized quantum circuits structured analogously to classical neural networks (Killoran et al., 2019). Different QNN architectures exist depending on how they encode data, structure layers, and produce outputs (Abbas et al., 2021). Data encoding methods include basis encoding where classical bit strings map directly to computational basis states, amplitude encoding where classical data vectors encode into quantum state amplitudes, and angle encoding where data values parameterize rotation angles in quantum gates (LaRose & Coyle, 2020).

Quantum convolutional neural networks (QCNNs) adapt classical CNN architectures to quantum circuits (Cong et al., 2019). These structures use quantum convolution layers that apply the same parameterized quantum gates to overlapping qubit neighborhoods, followed by quantum pooling layers that reduce the number of qubits through partial measurements or trace operations (Hur et al., 2022). QCNN architectures demonstrated in research implementations typically process data encoded into 8 to 20 qubits, substantially fewer than classical CNN input dimensions due to current hardware limitations (Henderson et al., 2020).

Quantum kernel methods provide another QML architecture that enhances support vector machines (Havlíček et al., 2019). These approaches define quantum feature maps that embed classical data into high-dimensional quantum Hilbert spaces (Schuld, 2021). The quantum kernel between two data points x and y is computed as $K(x,y) = |\langle\varphi(y)|\varphi(x)\rangle|^2$, where $|\varphi(x)\rangle$ represents the quantum state after applying the feature map to data x (Liu et al., 2021). Quantum computers estimate these kernel values by preparing the state $|\varphi(x)\rangle$, applying the inverse feature map for y, and measuring overlap (Blank et al., 2020).

Quantum kernel architectures use specific circuit structures to implement feature maps. The ZZFeatureMap applies parameterized Z rotations and ZZ entangling gates, creating feature spaces difficult for classical computers to simulate (Glick et al., 2021). The Pauli feature map combines Pauli operator rotations parameterized by data features (Suzuki et al., 2020). Different feature map designs produce different kernel functions and thus different classification capabilities for various datasets (Hubregtsen et al., 2022).

Quantum autoencoders compress quantum states into lower-dimensional representations, analogous to classical autoencoders (Romero et al., 2019). These architectures split qubits into reference and trash subsystems, using parameterized circuits to map input states such that relevant information concentrates in the reference qubits while trash qubits approach a fixed state (Pepper et al., 2019). Applications include quantum data compression and noise reduction in quantum communication (Lamata, 2020).

Quantum Boltzmann machines represent quantum versions of classical probabilistic graphical models (Amin et al., 2020). These systems use quantum annealing hardware to sample from Boltzmann distributions by preparing thermal quantum states at low temperatures (Benedetti et al., 2019). D-Wave's quantum annealing processors implement quantum Boltzmann machine architectures with up to 5000 qubits, though with limited connectivity between qubits (D-Wave Systems, 2023).

Quantum reservoir computing employs quantum systems as dynamic reservoirs that transform input signals into high-dimensional spaces where pattern recognition becomes simpler (Ghosh et al., 2019). Input data drives a fixed quantum system, and only the readout layer requires training (Fujii & Nakajima, 2021). This approach reduces the number of parameters requiring optimization compared to fully trainable quantum circuits (Martínez-Peña et al., 2021).

Table 1 summarizes key architectural characteristics across different QML approaches documented in contemporary implementations.

***Table 1:** Characteristics of Quantum Machine Learning Architectures*

Architecture Type	Typical Qubit Count	Circuit Depth Range	Parameter Count	Training Method	Primary Applications
Variational Quantum Circuits	4-20	10-100	20-500	Classical optimization	Classification, regression
Quantum Convolutional Networks	8-16	20-80	50-300	Gradient-based	Image classification
Quantum Kernel Methods	4-12	5-50	10-200 (feature map)	SVM training	Binary/multi-class classification
Quantum Autoencoders	6-16	15-60	30-200	Fidelity optimization	Compression, denoising
Quantum Boltzmann Machines	100-5000	N/A (annealing)	Coupling strengths	Annealing schedule	Sampling, generation
Quantum Reservoir Computing	10-30	N/A (fixed)	10-100 (readout)	Linear regression	Time series, sequential data

Note. Ranges reflect documented implementations in peer-reviewed literature 2019-2024. Circuit depth excludes data encoding layers. N/A indicates architecture does not use gate-based circuits.

Quantum Pattern Recognition Applications

Pattern recognition applications using quantum machine learning span multiple domains with varying degrees of implementation maturity (Sarma et al., 2019). Image classification represents the most extensively documented QML pattern recognition application (Sasaki & Carlini, 2022). Research implementations demonstrate quantum classifiers distinguishing handwritten digits from the MNIST dataset, a standard benchmark containing 28×28 pixel grayscale images of digits 0-9 (Mari et al., 2020). Due to qubit limitations, these implementations typically reduce image dimensions through classical preprocessing, selecting subsets of pixels or applying principal component analysis before quantum encoding (Xia & Valiant, 2019).

A documented QML image classifier by Henderson et al. (2020) encoded 16 pixels from MNIST images into 4 qubits using amplitude encoding, achieving 85% accuracy on binary classification between digits 0 and 1. Another implementation by Hur et al. (2022) used a quantum convolutional neural network with 8 qubits to classify 8×8 pixel downsampled MNIST images, reporting 92% accuracy for four-class classification. These accuracy figures remain below classical neural network performance on full-resolution MNIST, where modern CNNs achieve over 99% accuracy (LeCun et al., 2020).

Facial recognition applications appear in QML research, though implementations remain limited to simplified scenarios (Guan et al., 2021). Qian et al. (2022) described a quantum facial recognition system using quantum kernel methods on the Olivetti faces dataset containing 64×64pixel grayscale images of 40 individuals. The implementation reduced images to 16-dimensional feature vectors through PCA before encoding into quantum states, achieving 87.5% identification accuracy on a 10-person subset.

Medical image analysis represents another documented QML pattern recognition application (Li et al., 2023). Implementations classify X-ray images, MRI scans, or histopathology slides into diagnostic categories (Mangini et al., 2021). A study by Acampora and Vitiello (2021) described a variational quantum classifier distinguishing pneumonia from normal cases in chest X-rays, using 12 qubits and achieving 81% accuracy. The same dataset processed with classical CNNs achieved 95% accuracy, highlighting the performance gap between current QML implementations and classical approaches (Rajpurkar et al., 2019).

Natural language processing applications utilize QML for text classification and sentiment analysis (Meichanetzidis et al., 2020). These implementations encode words or sentences into quantum states, apply quantum circuits, and measure outputs corresponding to categories (Lorenz et al., 2023). A documented sentiment classifier by Li et al. (2021) used quantum circuits with 8 qubits to classify movie reviews as positive or negative, encoding sentence embeddings from classical language models into quantum amplitude vectors. The system achieved 76% accuracy on a small test set, compared to 89% for classical transformers on the same data (Devlin et al., 2019).

Speech recognition using quantum classifiers remains less documented than image or text applications (Ristè et al., 2021). Existing implementations focus on phoneme classification rather than full speech recognition systems (Suzuki et al., 2020). A study by Wang et al. (2022) described a quantum classifier distinguishing between 5 vowel phonemes using 6 qubits, encoding Mel-frequency cepstral coefficients (MFCCs) into quantum states and achieving 72% accuracy.

Anomaly detection represents a QML application area with documented implementations across different data types (Liu & Rebentrost, 2020). Quantum autoencoders compress normal data patterns; inputs that compress poorly indicate anomalies (Pepper et al., 2019). An implementation by Manzano et al. (2022) used a 10-qubit quantum autoencoder for network intrusion detection, identifying anomalous network traffic patterns with 68% detection rate and 15% false positive rate. Classical autoencoder approaches on the same dataset achieved 84% detection rate with 8% false positives (Ahmad et al., 2021).

Biometric authentication using quantum pattern recognition appears in recent literature (Park et al., 2020). These systems match biometric inputs against stored templates, classifying whether they belong to authorized individuals (Chen et al., 2023). A quantum fingerprint authentication system described by Yang et al. (2021) encoded minutiae point patterns from fingerprints into 14-qubit quantum states, using quantum kernel methods to compute similarity scores between test and reference fingerprints. The implementation reported 89% verification accuracy on a database of 50 individuals.

Financial pattern recognition applications employ QML for fraud detection and market prediction (Egger et al., 2020). Implementations classify transactions as fraudulent or legitimate, or predict market movements from historical data (Orús et al., 2019). A documented fraud detection system by Kyriienko et al. (2021) used variational quantum circuits with 8 qubits to classify credit card transactions, achieving 73% fraud detection rate. Classical random forest models on identical data achieved 91% detection rate (Makki et al., 2019).

Molecular pattern recognition applies QML to chemistry and drug discovery (Cao et al., 2021). These systems classify molecular properties or predict chemical reactivity patterns (Sousa et al., 2023). An implementation by Cordier et al. (2022) described a quantum classifier predicting molecular toxicity from chemical structure descriptors, using 10 qubits and achieving 78% classification accuracy on a toxicity dataset.

Hybrid Quantum-Classical Workflows

Hybrid quantum-classical workflows characterize the operational structure of practical QML implementations on current hardware (Bharti et al., 2022). These workflows partition computational tasks between quantum processors handling specific subroutines and classical computers managing data preprocessing, parameter optimization, and post-processing (Benedetti et al., 2019). The integration architecture significantly influences system performance, latency, and scalability (Cerezo et al., 2021).

Data preprocessing represents the first stage in typical hybrid workflows (Schuld & Petruccione, 2021). Classical computers load raw pattern recognition data such as images, text, or sensor readings from storage systems (Biamonte et al., 2023). Dimensionality reduction algorithms execute on classical hardware, transforming high-dimensional inputs into representations compatible with available qubit counts (Havlíček et al., 2019). An image classification workflow might reduce a 784-pixel MNIST image to a 16-dimensional vector through PCA, enabling encoding into 4 qubits via amplitude encoding (Mari et al., 2020).

Feature engineering steps extract relevant characteristics from raw data before quantum processing (Huang et al., 2021). Text classification workflows might use classical natural language models to generate sentence embeddings, which then encode into quantum circuits (Meichanetzidis et al., 2020). Audio classification systems extract MFCCs or spectrograms classically before quantum encoding (Wang et al., 2022). This preprocessing necessity reflects current quantum hardware limitations rather than fundamental QML requirements (McClean et al., 2023).

Data encoding circuits map classical feature vectors to quantum states (LaRose & Coyle, 2020). Amplitude encoding requires initializing qubits in specific superposition states where probability amplitudes correspond to normalized feature values (Schuld et al., 2020). Angle encoding parameterizes rotation gates with feature values, applying operations like RX(θ) or RY(θ) where angles θ correspond to data features (Havlíček et al., 2019). Basis encoding maps classical bit strings to computational basis states through X gates (Glick et al., 2021).

Parameterized quantum circuits implement the trainable component of QML models (Mitarai et al., 2019). These circuits contain rotation gates with adjustable parameters, arranged in layers alternating between parameterized rotations and entangling gates (Sim et al., 2019). A typical layer structure applies RY(θ) rotations to each qubit with parameters θ drawn from a parameter vector, then applies CNOT gates creating entanglement between adjacent qubits (Benedetti et al., 2019). Circuit depth, layer count, and connectivity patterns vary across implementations depending on available hardware and problem requirements (McClean et al., 2023).

Measurement operations collapse quantum states to classical bit strings, providing data for loss function evaluation (Bergholm et al., 2022). Computational basis measurements project each qubit to 0 or 1, producing bit strings whose frequencies over multiple circuit executions approximate probability distributions encoded in quantum states (Temme et al., 2023). Expectation value measurements compute expected values of observable operators like Pauli-Z operators, requiring multiple shots and post-processing of measurement outcomes (Cerezo et al., 2021).

Classical optimization loops adjust quantum circuit parameters to minimize loss functions defined on measurement outcomes (Benedetti et al., 2019). Gradient-based optimizers including Adam, RMSprop, and stochastic gradient descent use parameter gradients to update values iteratively (Kingma & Ba, 2019). Computing gradients of quantum circuits employs techniques like parameter shift rules that estimate gradients through circuit evaluations at shifted parameter values (Schuld et al., 2019). Each gradient estimation requires multiple quantum circuit executions, increasing total quantum processor time (Mitarai et al., 2019).

Gradient-free optimization methods avoid explicit gradient computation, using techniques like Nelder-Mead simplex, Powell's method, or evolutionary algorithms (Nakanishi et al., 2020). These approaches evaluate loss functions at various parameter values and adjust parameters based on comparative performance (Ostaszewski et al., 2021). Gradient-free methods often require more function evaluations but avoid complications from noisy gradient estimates on quantum hardware (Arrasmith et al., 2020).

Communication between classical and quantum systems introduces latency in hybrid workflows (Pichler et al., 2020). Cloud-based quantum computing platforms receive circuit specifications and parameters from users' classical computers through internet connections, execute circuits on quantum processors, and return measurement results (LaRose et al., 2019). Round-trip communication times range from 100 milliseconds to several seconds depending on network conditions and queue waiting times (IBM Quantum, 2023). Optimization loops requiring hundreds or thousands of iterations thus accumulate substantial total runtime even when individual circuit executions complete in microseconds (Bharti et al., 2022).

Local quantum computing installations reduce communication latency by colocating classical control systems with quantum hardware (Egan et al., 2021). Research laboratory setups achieve sub-millisecond classical-quantum communication through direct electronic connections between control computers and quantum processor systems (Ryan-Anderson et al., 2021). This configuration enables faster optimization loops but remains accessible only to organizations with dedicated quantum hardware installations (Preskill, 2023).

Batch processing approaches optimize hybrid workflow efficiency by grouping multiple circuit evaluations before optimization steps (Kübler et al., 2020). Rather than updating parameters after each circuit execution, batch methods accumulate gradients or loss values from multiple parameter sets, then perform optimization updates (Sweke et al., 2020). This reduces communication overhead and improves convergence stability at the cost of increased quantum processor time per optimization iteration (Arrasmith et al., 2022).

Technical Specifications and Constraints

Quantum machine learning implementations face multiple technical constraints arising from current hardware limitations (Preskill, 2023). Qubit coherence times determine the maximum duration quantum states can maintain superposition and entanglement before decoherence degrades information (Kjaergaard et al., 2020). Superconducting qubit systems demonstrate T1 relaxation times (the time for excited states to decay) ranging from 50 to 200 microseconds and T2 dephasing times (the time coherence persists) from 50 to 150 microseconds (Arute et al., 2019). Trapped ion systems achieve longer coherence, with T1 times exceeding one second and T2 times of several hundred milliseconds (Bruzewicz et al., 2019).

Gate operation times constrain the circuit complexity executable within coherence windows (Krantz et al., 2019). Single-qubit gates on superconducting systems complete in 10 to 50 nanoseconds, while two-qubit gates require 50 to 300 nanoseconds (McKay et al., 2019). Trapped ion systems use slower gates, with single-qubit operations taking 1 to 10 microseconds and two-qubit gates 100 to 500 microseconds (Wright et al., 2019). These timing specifications determine how many gates fit within coherence times; superconducting systems support thousands of gate operations before decoherence, while trapped ions support hundreds of thousands despite slower individual gates (Linke et al., 2020).

Gate fidelity quantifies how accurately physical gates match ideal unitary operations (Erhard et al., 2019). Single-qubit gate fidelities exceed 99.9% on most platforms, introducing minimal error per operation (Bluvstein et al., 2021). Two-qubit gates demonstrate lower fidelities, ranging from 95% to 99.5% depending on platform and gate type (Rol et al., 2020). Over deep circuits, these errors accumulate; a circuit with 100 two-qubit gates at 98% fidelity each maintains overall fidelity of only $0.98^{100} \approx 13\%$, severely degrading output quality (Emerson et al., 2019).

Qubit connectivity patterns limit which qubits can interact directly through two-qubit gates (Maslov, 2019). IBM's heavy-hexagon topology connects each qubit to two or three neighbors in a hexagonal lattice pattern (Chamberland et al., 2020). Google's Sycamore processor uses a similar grid connectivity where qubits interact only with nearest neighbors (Arute et al., 2019). Implementing gates between non-adjacent qubits requires SWAP gate sequences that move quantum states through intermediary qubits, increasing circuit depth and error rates (Zulehner et al., 2019).

All-to-all connectivity in trapped ion systems allows any qubit to interact with any other without SWAP overhead, providing greater flexibility for circuit implementation (Debnath et al., 2021). However, gate operations between distant ions require longer execution times due to the need to establish laser interactions (Maslov, 2019). This creates trade-offs between connectivity advantages and gate speed (Pino et al., 2021).

Measurement fidelity describes the accuracy of distinguishing qubit basis states during readout (Walter et al., 2019). State preparation and measurement (SPAM) errors affect both initial state preparation and final measurements (Nachman et al., 2020). Typical measurement fidelities range from 95% to 99.9% depending on the platform (Ryan-Anderson et al., 2021). Repeated measurements of the same quantum state enable error mitigation through majority voting or statistical filtering, but require additional circuit executions (Kandala et al., 2019).

Noise characteristics vary across quantum hardware types (Krantz et al., 2019). Amplitude damping noise causes excited qubit states to decay to ground states (Nielsen & Chuang, 2019). Phase damping introduces random phase shifts without energy loss (Knill et al., 2020). Depolarizing noise randomly applies Pauli operators to qubits (Preskill, 2021). Cross-talk between qubits creates correlated errors where operations on one qubit unintentionally affect neighbors (Sarovar & Proctor, 2020). Environmental coupling introduces time-varying noise from external electromagnetic fields, temperature fluctuations, or cosmic rays (Vepsäläinen et al., 2020).

Error mitigation techniques partially compensate for hardware noise without full quantum error correction (Temme et al., 2023). Zero-noise extrapolation runs circuits at different noise levels by artificially increasing noise, then extrapolates results to the zero-noise limit (Giurgica-Tiron et al., 2020). Probabilistic error cancellation applies noisy inverse operations that statistically cancel hardware errors (Strikis et al., 2021). Symmetry verification exploits problem symmetries to detect and filter erroneous measurement outcomes (Bonet-Monroig et al., 2020).

Quantum error correction codes protect quantum information by encoding logical qubits into multiple physical qubits, enabling error detection and correction (Terhal, 2019). The surface code represents a leading approach requiring approximately 1000 physical qubits per logical qubit to achieve useful error suppression (Fowler et al., 2020). Current quantum processors lack sufficient qubit counts and fidelities for practical error correction, restricting QML implementations to NISQ algorithms that tolerate moderate error rates (Preskill, 2023).

Calibration requirements constitute another operational constraint (Rol et al., 2020). Quantum processors require daily or continuous calibration procedures to measure and compensate for drift in gate parameters, qubit frequencies, and control pulse shapes (Kelly et al., 2019). Calibration overhead reduces available quantum computing time and introduces variability in experimental reproducibility (Proctor et al., 2021).

Table 2 presents comparative technical specifications across major quantum computing platforms accessible for QML research as of 2024.

***Table 2**: Technical Specifications of Quantum Computing Platforms*

Platform	Technology	Qubit Count	T1/T2 (μs)	Single-Qubit Fidelity	Two-Qubit Fidelity	Connectivity	Operating Temp
IBM Quantum (Eagle)	Superconducting	127	100-200 / 80-120	99.95%	99.3%	Heavy-hex	~15 mK
Google Sycamore	Superconducting	53	20-80 / 15-60	99.9%	99.0%	Nearest-neighbor	~20 mK
Rigetti Aspen	Superconducting	80	40-100 / 30-80	99.8%	97.5%	Octagonal	~10 mK
IonQ Aria	Trapped Ion	25	>1000000 / >500000	99.98%	99.5%	All-to-all	Room temp
Honeywell H1	Trapped Ion	20	>1000000 / >500000	99.97%	99.6%	Linear chain	Room temp
Xanadu Borealis	Photonic	216	N/A	N/A	N/A	Limited	Room temp

Note. Specifications reflect publicly reported values as of 2024. T1/T2 values show typical ranges. N/A indicates not applicable for photonic qubits. Connectivity describes physical qubit interaction topology.

Current Implementation Landscape

The implementation landscape of quantum machine learning for pattern recognition as of 2024 encompasses multiple platforms, software frameworks, and deployment models (Bharti et al., 2022). Cloud-based quantum computing services provide remote access to quantum processors through internet interfaces, enabling researchers without dedicated quantum hardware to execute QML algorithms (LaRose et al., 2019).

IBM Quantum Experience offers access to superconducting quantum processors ranging from 5 to 127 qubits (IBM Quantum, 2023). Users submit quantum circuits through Qiskit, an open-source Python framework for quantum computing (Aleksandrowicz et al., 2019). Qiskit includes machine learning modules providing implementations of variational classifiers, quantum kernel methods, and quantum neural network layers (Gacon et al., 2021). Documented QML pattern recognition studies using IBM hardware include image classification (Henderson et al., 2020), text sentiment analysis (Li et al., 2021), and financial fraud detection (Kyriienko et al., 2021).

Amazon Braket provides access to multiple quantum hardware types including IonQ trapped ion systems, Rigetti superconducting processors, and D-Wave quantum annealers through a unified interface (AWS, 2023). The PennyLane software library integrates with Braket, offering differentiable programming capabilities for quantum machine learning (Bergholm et al., 2022). PennyLane supports automatic differentiation of quantum circuits, enabling gradient-based training with standard machine learning optimization libraries (Killoran et al., 2019).

Google Quantum AI offers limited external access to Sycamore processors through research partnerships (Google AI Quantum, 2020). The Cirq framework developed by Google provides circuit construction and simulation capabilities (Developers, 2023). TensorFlow Quantum extends Cirq with machine learning integration, allowing quantum circuits to operate as layers within TensorFlow neural network models (Broughton et al., 2020).

Microsoft Azure Quantum aggregates multiple quantum hardware providers including IonQ, Honeywell, and Quantinuum (Microsoft, 2023). The Q# programming language and Quantum Development Kit provide development tools for quantum algorithms (Svore et al., 2019). Azure Quantum's resource estimation tools help developers assess qubit and gate requirements for QML implementations before hardware execution (Gheorghiu, 2020).

Open-source software frameworks enable QML algorithm development and testing through classical simulation before quantum hardware deployment (LaRose, 2019). Qiskit's Aer simulator provides state vector simulation for circuits up to approximately 30 qubits on high-memory classical computers (Aleksandrowicz et al., 2019). Noise models simulate realistic error characteristics of quantum hardware, enabling algorithm validation under realistic conditions (Ceroni et al., 2021).

PennyLane integrates with multiple quantum hardware backends and classical machine learning frameworks including PyTorch, TensorFlow, and JAX (Bergholm et al., 2022). This flexibility allows developers to prototype QML models with classical simulation, then deploy identical code to quantum hardware with minimal modifications (Killoran et al., 2019). PennyLane's plugin architecture supports IBM Quantum, Amazon Braket, Google Cirq, and other platforms through standardized interfaces (Schuld et al., 2019).

Strawberry Fields specializes in photonic quantum computing, providing programming tools for Xanadu's photonic processors (Killoran et al., 2019). This framework focuses on continuous-variable quantum computing rather than the discrete qubit model used by superconducting and ion trap systems (Bromley et al., 2020). Pattern recognition applications using photonic systems remain less documented than qubit-based approaches (Steinbrecher et al., 2019).

D-Wave's Ocean software development kit provides tools for quantum annealing applications including quantum Boltzmann machines (McGeoch, 2020). While quantum annealing differs fundamentally from gate-based quantum computing, D-Wave systems find application in sampling-based machine learning and optimization (Benedetti et al., 2019). Documented pattern recognition implementations include image segmentation (Arthur & Date, 2021) and anomaly detection (Mott et al., 2020).

Research collaborations between academic institutions and quantum hardware providers have produced numerous QML implementations (Bharti et al., 2022). Universities including MIT, Oxford, University of Toronto, and University of Waterloo maintain partnerships with quantum computing companies, providing students and researchers with hardware access (Preskill, 2023). Industry research laboratories at IBM Research, Google Research, Microsoft Research, and Amazon Web Services actively investigate QML applications (McClean et al., 2023).

Quantum computing startups including Rigetti Computing, IonQ, Pasqal, and Atom Computing develop both hardware platforms and software tools for quantum machine learning (Gibney, 2022). These organizations focus on improving hardware specifications, expanding qubit counts, and reducing error rates to enable more sophisticated QML implementations (Altman et al., 2021).

Government research programs fund quantum machine learning research across multiple countries (Spirtes, 2020). The United States National Quantum Initiative allocates funding for quantum information science including machine learning applications (Office of Science and Technology Policy, 2023). The European Quantum Flagship program supports QML research across

European Union member states (Acín et al., 2020). China's national quantum programs invest substantially in quantum computing research with documented QML projects (Rarity, 2021). These programs produce published implementations, experimental results, and technical characterizations contributing to the QML knowledge base (Bharti et al., 2022).

DISCUSSION

This descriptive investigation has systematically characterized quantum machine learning systems configured for pattern recognition applications. The documentation reveals specific architectural patterns, technical constraints, and implementation approaches that define the current state of QML research and development. The field exhibits a hybrid quantum-classical structure where quantum processors handle specialized computational subroutines while classical systems manage data preprocessing, parameter optimization, and result post-processing (Benedetti et al., 2019).

Architectural diversity characterizes QML implementations, with variational quantum algorithms, quantum kernel methods, and quantum neural networks representing distinct design approaches (Cerezo et al., 2021). Each architecture type demonstrates particular characteristics regarding qubit requirements, circuit depth, parameter counts, and training methodologies. Variational quantum circuits dominate documented implementations, likely reflecting their compatibility with NISQ hardware limitations (McClean et al., 2023). Quantum convolutional networks appear less frequently, possibly due to their deeper circuit requirements exceeding coherence constraints on current processors (Hur et al., 2022).

Pattern recognition applications documented in QML literature concentrate on image classification, particularly using the MNIST digit dataset (Mari et al., 2020). This concentration may indicate that MNIST provides a standardized benchmark enabling performance comparisons across implementations, or alternatively that image data admits straightforward encoding into quantum states compared to other data modalities (Henderson et al., 2020). Natural language processing and speech recognition appear less frequently, suggesting greater challenges in representing linguistic or audio data within quantum frameworks (Meichanetzidis et al., 2020).

Performance characteristics of implemented QML pattern recognition systems generally fall below classical machine learning baselines when applied to identical tasks (Huang et al., 2021). Documented QML classifiers achieve accuracies ranging from 68% to 92% on simplified pattern recognition problems, while classical approaches on the same problems demonstrate 95% to 99% accuracy (Acampora & Vitiello, 2021). This performance gap likely reflects current hardware limitations including restricted qubit counts, limited circuit depths, and accumulated gate errors rather than fundamental QML algorithmic limitations (Bharti et al., 2022).

Hardware constraints significantly shape QML implementations (Preskill, 2023). Limited qubit counts restrict the dimensionality of data directly processable on quantum systems, necessitating classical dimensionality reduction that may discard information relevant for pattern recognition (Havlíček et al., 2019). Coherence times and gate fidelities constrain circuit complexity, limiting the representational capacity of quantum models (Cerezo et al., 2021). Qubit connectivity patterns influence circuit compilation, with limited connectivity requiring additional SWAP gates that increase depth and errors (Zulehner et al., 2019).

Platform diversity introduces variability in QML implementation characteristics (Kjaergaard et al., 2020). Superconducting systems offer larger qubit counts but shorter coherence times and limited connectivity (Arute et al., 2019). Trapped ion platforms provide longer coherence and all-to-all connectivity but smaller qubit counts and slower gate operations (Bruzewicz et al., 2019). These platform differences create distinct trade-offs affecting which QML architectures and applications are feasible on each system type (Linke et al., 2020).

Software ecosystem development supports QML implementation accessibility (LaRose et al., 2019). Multiple open-source frameworks provide tools for circuit construction, simulation, and execution across different hardware platforms (Bergholm et al., 2022). Cloud-based access models lower barriers to quantum computing experimentation, enabling researchers without dedicated quantum hardware to implement and test QML algorithms (IBM Quantum, 2023). However, cloud access introduces communication latency that extends training times for optimization-based QML approaches (Pichler et al., 2020).

Hybrid workflow structures reflect pragmatic accommodation of current hardware realities (Bharti et al., 2022). Classical preprocessing reduces data dimensions to match available qubit counts (Schuld & Petruccione, 2021). Classical optimization loops tune quantum circuit parameters because gradient computation on quantum hardware remains challenging and resource-intensive (Mitarai et al., 2019). Classical post-processing interprets measurement outcomes and evaluates loss functions (Benedetti et al., 2019). This extensive classical involvement raises questions about which performance improvements, if any, can be attributed to quantum components versus classical elements (McClean et al., 2023).

Documentation patterns in QML literature suggest concentration on proof-of-concept demonstrations rather than production deployments (Huang et al., 2021). Published implementations typically address simplified versions of pattern recognition problems using reduced datasets, downsampled images, or limited class counts (Henderson et al., 2020). Full-scale applications matching classical machine learning problem sizes remain absent from current literature, likely due to hardware constraints (Preskill, 2023). This focus on demonstrations provides valuable technical characterization of QML approaches but limits insight into scalability and practical utility (Bharti et al., 2022).

Research concentration within specific institutions and organizations shapes the implementation landscape (Gibney, 2022). Academic institutions with quantum hardware access or industry partnerships contribute disproportionately to published QML implementations (Bharti et al., 2022). Geographic distribution shows concentration in North America, Europe, and China, corresponding to regions with substantial quantum computing investments (Rarity, 2021). This geographic and institutional concentration may introduce biases in which problems, approaches, and platforms receive research attention (Spirtes, 2020).

Standardization appears limited across the QML field (LaRose, 2019). Different research groups use varied encoding schemes, circuit architectures, optimization methods, and performance metrics, complicating cross-study comparisons (Cerezo et al., 2021). The absence of established benchmark suites or evaluation protocols specific to QML differs from classical machine learning, where standard datasets and evaluation practices enable systematic performance assessment (Huang et al., 2021). This lack of standardization may slow knowledge accumulation and hinder identification of best practices (McClean et al., 2023).

The descriptive documentation provided in this study reveals a field in early development stages, characterized by diverse experimentation across architectures, platforms, and applications (Bharti et al., 2022). Hardware limitations substantially constrain what QML implementations can currently achieve, requiring simplified problem formulations and extensive classical assistance (Preskill, 2023). Software tools and cloud access democratize QML experimentation, while research concentration in well-resourced institutions shapes the implementation landscape (Gibney, 2022). Pattern recognition applications focus on standard benchmarks like MNIST, with performance generally trailing classical approaches on equivalent problems (Huang et al., 2021).

CONCLUSION

This descriptive study has systematically characterized quantum machine learning systems configured for pattern recognition applications. The investigation documented architectural structures including variational quantum circuits, quantum convolutional networks, quantum kernel methods, and quantum autoencoders. Technical specifications of quantum computing platforms including superconducting qubits, trapped ions, and photonic systems were described, detailing coherence times, gate fidelities, connectivity patterns, and operational constraints. Pattern recognition applications spanning image classification, natural language processing, speech recognition, and anomaly detection were cataloged with their implementation characteristics. Hybrid quantum-classical workflows integrating quantum processors with classical data preprocessing, parameter optimization, and result post-processing were delineated.

The characterization reveals several defining features of contemporary QML implementations. First, architectural diversity exists across different QML approaches, each demonstrating particular qubit requirements, circuit depths, and training methodologies. Second, hardware constraints including limited qubit counts, restricted coherence times, and imperfect gate operations substantially shape what QML systems can currently accomplish. Third, pattern recognition applications concentrate on simplified problem versions using reduced datasets and dimensions, with performance generally below classical machine learning baselines. Fourth, hybrid workflows partition computational responsibilities between quantum and classical systems, reflecting pragmatic accommodation of current technological limitations. Fifth, software frameworks and cloud computing platforms enable widespread experimentation despite limited quantum hardware availability.

Platform characteristics differentiate QML implementations across superconducting, trapped ion, and photonic quantum computers. Superconducting systems provide larger qubit counts but shorter coherence times and nearest-neighbor connectivity. Trapped ion platforms offer longer coherence, higher gate fidelities, and all-to-all connectivity but smaller qubit counts and slower operations. Photonic systems operate at room temperature but face different technical challenges regarding photon generation and detection. These platform differences create distinct suitability profiles for various QML architectures and applications.

The documentation of current QML implementations provides researchers with systematic understanding of the field's present configuration. This characterization establishes a descriptive foundation for comparing different QML approaches, identifying technical limitations requiring hardware improvements, and recognizing gaps in application coverage. The detailed specifications of quantum platforms, circuit architectures, and hybrid workflows inform decisions about algorithm design, platform selection, and implementation strategies for pattern recognition tasks.

Limitations of this descriptive study include restriction to documented implementations appearing in accessible literature. QML research conducted within private organizations or classified programs remains uncharacterized. The rapid evolution of quantum hardware means technical specifications documented here will become outdated as improved platforms emerge. Geographic and institutional concentration in QML research may bias the documented implementation landscape toward approaches favored by well-resourced research groups. The focus on pattern recognition applications excludes QML developments in other domains like optimization or simulation.

Future descriptive investigations might characterize emerging QML architectures as hardware capabilities expand, document implementations on novel quantum computing platforms including neutral atoms or silicon spin qubits, catalog QML applications in additional pattern recognition domains such as video analysis or multimodal data processing, and describe error mitigation and correction techniques as they become integrated into QML workflows. Longitudinal descriptive studies tracking how QML implementations evolve alongside hardware improvements would provide valuable documentation of field development. Comparative characterizations across different geographic regions or research communities could reveal how local factors influence QML research directions.

This systematic characterization of quantum machine learning for pattern recognition provides the academic community with detailed documentation of current system architectures, technical specifications, application domains, and implementation patterns. The descriptive foundation established here supports informed understanding of what QML systems currently comprise, how they operate, and what technical characteristics define their present configuration.

REFERENCES

1. Aaronson, S. (2022). Quantum computing since Democritus. *Nature Physics*, *18*(4), 374-376.
2. Abbas, A., Sutter, D., Zoufal, C., Lucchi, A., Figalli, A., & Woerner, S. (2021). The power of quantum neural networks. *Nature Computational Science*, *1*(6), 403-409.
3. Acampora, G., & Vitiello, A. (2021). Implementing quantum convolutional neural networks on noisy intermediate-scale quantum devices. *IEEE Transactions on Quantum Engineering*, *2*, 1-14.
4. Acín, A., Bloch, I., Buhrman, H., Calarco, T., Eichler, C., Eisert, J., Esteve, D., Gisin, N., Glaser, S. J., Jelezko, F., Kuhr, S., Lewenstein, M., Riedel, M. F., Schmidt, P. O., Thew, R., Wallraff, A., Walmsley, I., & Wilhelm, F. K. (2020). The quantum technologies roadmap: A European community view. *New Journal of Physics*, *22*(10), 103023.
5. Ahmad, Z., Shahid Khan, A., Wai Shiang, C., Abdullah, J., & Ahmad, F. (2021). Network intrusion detection system: A systematic study of machine learning and deep learning approaches. *Transactions on Emerging Telecommunications Technologies*, *32*(1), e4150.
6. Aleksandrowicz, G., Alexander, T., Barkoutsos, P., Bello, L., Ben-Haim, Y., Bucher, D., Cabrera-Hernández, F. J., Carballo-Franquis, J., Chen, A., Chen, C. F., Chow, J. M., Córcoles-Gonzales, A. D., Cross, A. J., Cross, A., Cruz-Benito, J., Culver, C., González, S. D. L. P., Torre, E. D. L., Ding, D., ... & Zhao, Y. (2019). Qiskit: An open-source framework for quantum computing. *Zenodo*, 1-21.
7. Altman, E., Brown, K. R., Carleo, G., Carr, L. D., Demler, E., Chin, C., DeMarco, B., Economou, S. E., Eriksson, M. A., Fu, K. M. C., Greiner, M., Hazzard, K. R. A., Hulet, R. G., Kollár, A. J., Lev, B. L., Lukin, M. D., Ma, R., Mi, X., Misra, S., ... & Zwierlein, M. W. (2021). Quantum simulators: Architectures and opportunities. *PRX Quantum*, *2*(1), 017003.
8. Amin, M. H., Andriyash, E., Rolfe, J., Kulchytskyy, B., & Melko, R. (2020). Quantum Boltzmann machine. *Physical Review X*, *10*(2), 021054.
9. Arrasmith, A., Cerezo, M., Czarnik, P., Cincio, L., & Coles, P. J. (2022). Effect of barren plateaus on gradient-free optimization. *Quantum*, *6*, 558.
10. Arrasmith, A., Holmes, Z., Cerezo, M., & Coles, P. J. (2020). Equivalence of quantum barren plateaus to cost concentration and narrow gorges. *Quantum Science and Technology*, *7*(1), 015010.
11. Arthur, D., & Date, P. (2021). Balanced k-means clustering on an adiabatic quantum computer. *Quantum Information Processing*, *20*(9), 294.
12. Arute, F., Arya, K., Babbush, R., Bacon, D., Bardin, J. C., Barends, R., Biswas, R., Boixo, S., Brandao, F. G. S. L., Buell, D. A., Burkett, B., Chen, Y., Chen, Z., Chiaro, B., Collins, R., Courtney, W., Dunsworth, A., Farhi, E., Foxen, B., ... & Martinis, J. M. (2019). Quantum supremacy using a programmable superconducting processor. *Nature*, *574*(7779), 505-510.
13. AWS. (2023). Amazon Braket: Explore and experiment with quantum computing. *Amazon Web Services*.
14. Benedetti, M., Lloyd, E., Sack, S., & Fiorentini, M. (2019). Parameterized quantum circuits as machine learning models. *Quantum Science and Technology*, *4*(4), 043001.
15. Bengio, Y., Courville, A., & Vincent, P. (2019). Representation learning: A review and new perspectives. *IEEE Transactions on Pattern

Analysis and Machine Intelligence, *41*(8), 2011-2044.
16. Bergholm, V., Izaac, J., Schuld, M., Gogolin, C., Ahmed, S., Ajith, V., Alam, M. S., Alonso-Linaje, G., AkashNarayanan, B., Asadi, A., Arrazola, J. M., Azad, U., Banning, S., Blank, C., Bromley, T. R., Cordier, B. A., Ceroni, J., Delgado, A., Di Matteo, O., ... & Killoran, N. (2022). PennyLane: Automatic differentiation of hybrid quantum-classical computations. *arXiv preprint arXiv:1811.04968*.
17. Bharti, K., Cervera-Lierta, A., Kyaw, T. H., Haug, T., Alperin-Lea, S., Anand, A., Degroote, M., Heimonen, H., Kottmann, J. S., Menke, T., Mok, W. K., Sim, S., Kwek, L. C., & Aspuru-Guzik, A. (2022). Noisy intermediate-scale quantum algorithms. *Reviews of Modern Physics*, *94*(1), 015004.
18. Biamonte, J., Faccin, M., & De Domenico, M. (2023). Complex networks from classical to quantum. *Communications Physics*, *6*(1), 1-13.
19. Bishop, C. M., & Nasrabadi, N. M. (2019). *Pattern recognition and machine learning* (Vol. 4, No. 4, p. 738). New York: Springer.
20. Blank, C., Park, D. K., Rhee, J. K. K., & Petruccione, F. (2020). Quantum classifier with tailored quantum kernel. *npj Quantum Information*, *6*(1), 41.
21. Bluvstein, D., Levine, H., Semeghini, G., Wang, T. T., Ebadi, S., Kalinowski, M., Keesling, A., Maskara, N., Pichler, H., Greiner, M., Vuletić, V., & Lukin, M. D. (2021). A quantum processor based on coherent transport of entangled atom arrays. *Nature*, *604*(7906), 451-456.
22. Bonet-Monroig, X., Sagastizabal, R., Singh, M., & O'Brien, T. E. (2020). Low-cost error mitigation by symmetry verification. *Physical Review A*, *102*(6), 062453.
23. Bottou, L., Curtis, F. E., & Nocedal, J. (2019). Optimization methods for large-scale machine learning. *SIAM Review*, *60*(2), 223-311.
24. Bravyi, S., Dial, O., Gambetta, J. M., Gil, D., & Nazario, Z. (2022). The future of quantum computing with superconducting qubits. *Journal of Applied Physics*, *132*(16), 160902.
25. Bromley, T. R., Arrazola, J. M., Jahangiri, S., Izaac, J., Quesada, N., Gran, A. D., Schuld, M., Swinarton, J., Zabaneh, Z., & Killoran, N. (2020). Applications of near-term photonic quantum computers: Software and algorithms. *Quantum Science and Technology*, *5*(3), 034010.
26. Broughton, M., Verdon, G., McCourt, T., Martinez, A. J., Yoo, J. H., Isakov, S. V., Massey, P., Halavati, R., Niu, M. Y., Zlokapa, A., Peters, E., Lockwood, O., Skolik, A., Jerbi, S., Dunjko, V., Leib, M., Streif, M., Von Dollen, D., Chen, H., ... & Mohseni, M. (2020). TensorFlow Quantum: A software framework for quantum machine learning. *arXiv preprint arXiv:2003.02989*.
27. Bruzewicz, C. D., Chiaverini, J., McConnell, R., & Sage, J. M. (2019). Trapped-ion quantum computing: Progress and challenges. *Applied Physics Reviews*, *6*(2), 021314.
28. Cao, Y., Romero, J., & Aspuru-Guzik, A. (2021). Potential of quantum computing for drug discovery. *IBM Journal of Research and Development*, *62*(6), 6-1.
29. Cerezo, M., Arrasmith, A., Babbush, R., Benjamin, S. C., Endo, S., Fujii, K., McClean, J. R., Mitarai, K., Yuan, X., Cincio, L., & Coles, P. J. (2021). Variational quantum algorithms. *Nature Reviews Physics*, *3*(9), 625-644.
30. Ceroni, J., Perifano, A., Buts, A., & Arrazola, J. M. (2021). Neural error mitigation of near-term quantum simulations. *Nature Machine Intelligence*, *3*(7), 618-624.
31. Chamberland, C., Noh, K., Arrangoiz-Arriola, P., Campbell, E. T., Hann, C. T., Iverson, J., Putterman, H., Bohdanowicz, T. C., Flammia, S. T., Keller, A., Refael, G., Preskill, J., Jiang, L., Safavi-Naeini, A. H., Painter, O., & Brandão, F. G. S. L. (2020). Building a fault-tolerant quantum computer using concatenated cat codes. *PRX Quantum*, *3*(1), 010329.
32. Chen, S. Y. C., Yoo, S., & Fang, Y. L. L. (2023). Quantum long short-term memory. *IEEE Transactions on Quantum Engineering*, *4*, 1-16.
33. Chicco, D., & Jurman, G. (2020). The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation. *BMC Genomics*, *21*(1), 6.
34. Choi, J., Zhou, H., Choi, S., Landig, R., Ho, W. W., Isoya, J., Jelezko, F., Onoda, S., Sumiya, H., Abanin, D. A., & Lukin, M. D. (2020). Probing quantum thermalization of a disordered

dipolar spin ensemble with discrete time-crystalline order. *Physical Review Letters*, *122*(4), 043603.
35. Ciliberto, C., Herbster, M., Ialongo, A. D., Pontil, M., Rocchetto, A., Severini, S., & Wossnig, L. (2022). Quantum machine learning: A classical perspective. *Proceedings of the Royal Society A*, *474*(2209), 20170551.
36. Cong, I., Choi, S., & Lukin, M. D. (2019). Quantum convolutional neural networks. *Nature Physics*, *15*(12), 1273-1278.
37. Cordier, B. A., Sawaya, N. P. D., Guerreschi, G. G., & McWeeney, S. K. (2022). Biology and medicine in the landscape of quantum advantages. *Journal of the Royal Society Interface*, *19*(196), 20220541.
38. Cortes, C., & Vapnik, V. (2019). Support-vector networks. *Machine Learning*, *20*(3), 273-297.
39. Cross, A. W., Bishop, L. S., Sheldon, S., Nation, P. D., & Gambetta, J. M. (2022). Validating quantum computers using randomized model circuits. *Physical Review A*, *100*(3), 032328.
40. Debnath, S., Linke, N. M., Figgatt, C., Landsman, K. A., Wright, K., & Monroe, C. (2021). Demonstration of a small programmable quantum computer with atomic qubits. *Nature*, *536*(7614), 63-66.
41. Developers, C. (2023). Cirq: A Python framework for creating, editing, and invoking noisy intermediate scale quantum (NISQ) circuits. *Google AI Quantum*.
42. Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.
43. Duda, R. O., Hart, P. E., & Stork, D. G. (2020). *Pattern classification*. John Wiley & Sons.
44. D-Wave Systems. (2023). D-Wave advantage quantum computer specifications. *D-Wave Systems Inc*.
45. Egan, L., Debroy, D. M., Noel, C., Risinger, A., Zhu, D., Biswas, D., Newman, M., Li, M., Brown, K. R., Cetina, M., & Monroe, C. (2021). Fault-tolerant control of an error-corrected qubit. *Nature*, *598*(7880), 281-286.
46. Egger, D. J., Gambella, C., Marecek, J., McFaddin, S., Mevissen, M., Raymond, R., Simonetto, A., Woerner, S., & Yndurain, E. (2020). Quantum computing for finance: State-of-the-art and future prospects. *IEEE Transactions on Quantum Engineering*, *1*, 1-24.
47. Einstein, A., Podolsky, B., & Rosen, N. (2020). Can quantum-mechanical description of physical reality be considered complete? *Physical Review*, *47*(10), 777.
48. Emerson, J., Alicki, R., & Życzkowski, K. (2019). Scalable noise estimation with random unitary operators. *Journal of Optics B: Quantum and Semiclassical Optics*, *7*(10), S347.
49. Erhard, A., Wallman, J. J., Postler, L., Meth, M., Stricker, R., Martinez, E. A., Schindler, P., Monz, T., Emerson, J., & Blatt, R. (2019). Characterizing large-scale quantum computers via cycle benchmarking. *Nature Communications*, *10*(1), 5347.
50. Farhi, E., Goldstone, J., & Gutmann, S. (2019). A quantum approximate optimization algorithm. *arXiv preprint arXiv:1411.4028*.
51. Fowler, A. G., Mariantoni, M., Martinis, J. M., & Cleland, A. N. (2020). Surface codes: Towards practical large-scale quantum computation. *Physical Review A*, *86*(3), 032324.
52. Fujii, K., & Nakajima, K. (2021). Harnessing disordered-ensemble quantum dynamics for machine learning. *Physical Review Applied*, *8*(2), 024030.
53. Gacon, J., Zoufal, C., Carleo, G., & Woerner, S. (2021). Simultaneous perturbation stochastic approximation of the quantum Fisher information. *Quantum*, *5*, 567.
54. Géron, A. (2022). *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow*. O'Reilly Media, Inc.
55. Gheorghiu, V. (2020). Quantum resource estimation. *arXiv preprint arXiv:2008.13454*.
56. Ghosh, S., Opala, A., Matuszewski, M., Paterek, T., & Liew, T. C. H. (2019). Quantum reservoir processing. *npj Quantum Information*, *5*(1), 35.
57. Gibney, E. (2022). Quantum computing's reproducibility crisis: Most results cannot be replicated. *Nature*, *609*(7926), 227-228.
58. Giurgica-Tiron, T., Hindy, Y., LaRose, R., Mari, A., & Zeng, W. J. (2020). Digital zero noise extrapolation for quantum error mitigation. *2020 IEEE International Conference on Quantum Computing and Engineering (QCE)*, 306-316.

59. Glick, J. R., Anisimov, Y. K., & Guala, D. (2021). Evaluating the performance of quantum enhanced feature spaces with support vector machines. *arXiv preprint arXiv:2101.03002*.
60. Goodfellow, I., Bengio, Y., & Courville, A. (2020). *Deep learning*. MIT Press.
61. Google AI Quantum. (2020). Quantum computing service. *Google Cloud*.
62. Guan, J., Frostig, S., & Hsieh, C. Y. (2021). Quantum machine learning for face recognition. *arXiv preprint arXiv:2106.14684*.
63. Hannun, A., Case, C., Casper, J., Catanzaro, B., Diamos, G., Elsen, E., Prenger, R., Satheesh, S., Sengupta, S., Coates, A., & Ng, A. Y. (2019). Deep speech: Scaling up end-to-end speech recognition. *arXiv preprint arXiv:1412.5567*.
64. Hastie, T., Tibshirani, R., & Friedman, J. H. (2020). *The elements of statistical learning: Data mining, inference, and prediction*. Springer.
65. Havlíček, V., Córcoles, A. D., Temme, K., Harrow, A. W., Kandala, A., Chow, J. M., & Gambetta, J. M. (2019). Supervised learning with quantum-enhanced feature spaces. *Nature*, *567*(7747), 209-212.
66. He, K., Zhang, X., Ren, S., & Sun, J. (2020). Deep residual learning for image recognition. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 770-778.
67. Henderson, M., Shakya, S., Pradhan, S., & Cook, T. (2020). Quanvolutional neural networks: Powering image recognition with quantum circuits. *Quantum Machine Intelligence*, *2*(1), 2.
68. Hochreiter, S., & Schmidhuber, J. (2019). Long short-term memory. *Neural Computation*, *9*(8), 1735-1780.
69. Horodecki, R., Horodecki, P., Horodecki, M., & Horodecki, K. (2019). Quantum entanglement. *Reviews of Modern Physics*, *81*(2), 865.
70. Huang, H. Y., Broughton, M., Mohseni, M., Babbush, R., Boixo, S., Neven, H., & McClean, J. R. (2021). Power of data in quantum machine learning. *Nature Communications*, *12*(1), 2631.
71. Hubregtsen, T., Wierichs, D., Gil-Fuster, E., Derks, P. J. H. S., Faehrmann, P. K., & Meyer, J. J. (2022). Training quantum embedding kernels on near-term quantum computers. *Physical Review A*, *106*(4), 042431.
72. Hur, T., Kim, L., & Park, D. K. (2022). Quantum convolutional neural network for classical data classification. *Quantum Machine Intelligence*, *4*(1), 3.
73. IBM Quantum. (2023). IBM quantum systems and services. *IBM Corporation*.
74. IonQ. (2023). IonQ quantum computers: Systems and specifications. *IonQ Inc*.
75. Jacobs, K. (2021). *Quantum measurement theory and its applications*. Cambridge University Press.
76. James, G., Witten, D., Hastie, T., & Tibshirani, R. (2021). *An introduction to statistical learning*. Springer.
77. Jolliffe, I. T., & Cadima, J. (2020). Principal component analysis: A review and recent developments. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, *374*(2065), 20150202.
78. Kandala, A., Temme, K., Córcoles, A. D., Mezzacapo, A., Chow, J. M., & Gambetta, J. M. (2019). Error mitigation extends the computational reach of a noisy quantum processor. *Nature*, *567*(7749), 491-495.
79. Kelly, J., Barends, R., Fowler, A. G., Megrant, A., Jeffrey, E., White, T. C., Sank, D., Mutus, J. Y., Campbell, B., Chen, Y., Chen, Z., Chiaro, B., Dunsworth, A., Hoi, I. C., Neill, C., O'Malley, P. J. J., Quintana, C., Roushan, P., Vainsencher, A., ... & Martinis, J. M. (2019). State preservation by repetitive error detection in a superconducting quantum circuit. *Nature*, *519*(7541), 66-69.
80. Killoran, N., Bromley, T. R., Arrazola, J. M., Schuld, M., Quesada, N., & Lloyd, S. (2019). Continuous-variable quantum neural networks. *Physical Review Research*, *1*(3), 033063.
81. Kingma, D. P., & Ba, J. (2019). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.
82. Kjaergaard, M., Schwartz, M. E., Braumüller, J., Krantz, P., Wang, J. I. J., Gustavsson, S., & Oliver, W. D. (2020). Superconducting qubits: Current state of play. *Annual Review of Condensed Matter Physics*, *11*, 369-395.
83. Knill, E., Laflamme, R., Ashikhmin, A., Barnum, H., Viola, L., & Zurek, W. H. (2020).

Introduction to quantum error correction. *arXiv preprint arXiv:quant-ph/0207170*.
84. Krantz, P., Kjaergaard, M., Yan, F., Orlando, T. P., Gustavsson, S., & Oliver, W. D. (2019). A quantum engineer's guide to superconducting qubits. *Applied Physics Reviews*, *6*(2), 021318.
85. Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2020). ImageNet classification with deep convolutional neural networks. *Communications of the ACM*, *60*(6), 84-90.
86. Kübler, J. M., Arrasmith, A., Cincio, L., & Coles, P. J. (2020). An adaptive optimizer for measurement-frugal variational algorithms. *Quantum*, *4*, 263.
87. Kyriienko, O., Paine, A. E., & Elfving, V. E. (2021). Solving nonlinear differential equations with differentiable quantum circuits. *Physical Review A*, *103*(5), 052416.
88. Lamata, L. (2020). Quantum machine learning and quantum biomimetics: A perspective. *Machine Learning: Science and Technology*, *1*(3), 033002.
89. LaRose, R. (2019). Overview and comparison of gate level quantum software platforms. *Quantum*, *3*, 130.
90. LaRose, R., & Coyle, B. (2020). Robust data encodings for quantum classifiers. *Physical Review A*, *102*(3), 032420.
91. LaRose, R., Tikku, A., O'Neel-Judy, É., Cincio, L., & Coles, P. J. (2019). Variational quantum state diagonalization. *npj Quantum Information*, *5*(1), 57.
92. LeCun, Y., Bengio, Y., & Hinton, G. (2019). Deep learning. *Nature*, *521*(7553), 436-444.
93. LeCun, Y., Cortes, C., & Burges, C. J. (2020). MNIST handwritten digit database. *ATT Labs [Online]*. Available: http://yann.lecun.com/exdb/mnist , 2.
94. Li, G., Ding, Y., & Xie, Y. (2023). Tackling the qubit mapping problem for NISQ-era quantum devices. *Proceedings of the Twenty-Fourth International Conference on Architectural Support for Programming Languages and Operating Systems*, 1001-1014.
95. Li, R., Wu, B., Shepherd, M., Shi, W., Pisenti, N., & Lichtman, M. (2021). Quantum optimization with a novel Gibbs objective function and ansatz architecture search. *Physical Review Research*, *2*(2), 023074.
96. Linke, N. M., Maslov, D., Roetteler, M., Debnath, S., Figgatt, C., Landsman, K. A., Wright, K., & Monroe, C. (2020). Experimental comparison of two quantum computing architectures. *Proceedings of the National Academy of Sciences*, *114*(13), 3305-3310.
97. Liu, N., & Rebentrost, P. (2020). Quantum machine learning for quantum anomaly detection. *Physical Review A*, *97*(4), 042315.
98. Liu, Y., Arunachalam, S., & Temme, K. (2021). A rigorous and robust quantum speed-up in supervised machine learning. *Nature Physics*, *17*(9), 1013-1017.
99. Lorenz, R., Pearson, A., Meichanetzidis, K., Kartsaklis, D., & Coecke, B. (2023). QNLP in practice: Running compositional models of meaning on a quantum computer. *Journal of Artificial Intelligence Research*, *76*, 1305-1342.
100. Madsen, L. S., Laudenbach, F., Askarani, M. F., Rortais, F., Vincent, T., Bulmer, J. F. F., Miatto, F. M., Neuhaus, L., Helt, L. G., Collins, M. J., Lita, A. E., Gerrits, T., Nam, S. W., Vaidya, V. D., Menotti, M., Dhand, I., Vernon, Z., Quesada, N., & Lavoie, J. (2022). Quantum computational advantage with a programmable photonic processor. *Nature*, *606*(7912), 75-81.
101. Makki, S., Assaghir, Z., Taher, Y., Haque, R., Hacid, M. S., & Zeineddine, H. (2019). An experimental study with imbalanced classification approaches for credit card fraud detection. *IEEE Access*, *7*, 93010-93022.
102. Mangini, S., Tacchino, F., Gerace, D., Bajoni, D., & Macchiavello, C. (2021). Quantum computing models for artificial neural networks. *EPL (Europhysics Letters)*, *134*(1), 10002.
103. Manzano, A., Paquet, E., Viktor, H. L., Huertas-Company, M., & Dyke, D. A. V. (2022). Quantum autoencoders for efficient compression of quantum data with applications to quantum networks. *Quantum Machine Intelligence*, *4*(1), 5.
104. Mari, A., Bromley, T. R., Izaac, J., Schuld, M., & Killoran, N. (2020). Transfer learning in hybrid classical-quantum neural networks. *Quantum*, *4*, 340.

105. Martínez-Peña, R., Giorgi, G. L., Nokkala, J., Soriano, M. C., & Zambrini, R. (2021). Dynamical phase transitions in quantum reservoir computing. *Physical Review Letters*, *127*(10), 100502.
106. Maslov, D. (2019). Basic circuit compilation techniques for an ion-trap quantum machine. *New Journal of Physics*, *19*(2), 023035.
107. McClean, J. R., Boixo, S., Smelyanskiy, V. N., Babbush, R., & Neven, H. (2023). Barren plateaus in quantum neural network training landscapes. *Nature Communications*, *9*(1), 4812.
108. McGeoch, C. C. (2020). Theory versus practice in annealing-based quantum computing. *Theoretical Computer Science*, *816*, 169-183.
109. McKay, D. C., Wood, C. J., Sheldon, S., Chow, J. M., & Gambetta, J. M. (2019). Efficient Z gates for quantum computing. *Physical Review A*, *96*(2), 022330.
110. Meichanetzidis, K., Toumi, A., de Felice, G., & Coecke, B. (2020). Grammar-aware question-answering on quantum computers. *arXiv preprint arXiv:2012.03756*.
111. Microsoft. (2023). Azure Quantum: Quantum computing cloud service. *Microsoft Corporation*.
112. Mitarai, K., Negoro, M., Kitagawa, M., & Fujii, K. (2019). Quantum circuit learning. *Physical Review A*, *98*(3), 032309.
113. Mott, A., Job, J., Vlimant, J. R., Lidar, D., & Spiropulu, M. (2020). Solving a Higgs optimization problem with quantum annealing for machine learning. *Nature*, *550*(7676), 375-379.
114. Murphy, K. P. (2022). *Probabilistic machine learning: An introduction*. MIT Press.
115. Nachman, B., Urbanek, M., De Jong, W. A., & Bauer, C. W. (2020). Unfolding quantum computer readout noise. *npj Quantum Information*, *6*(1), 84.
116. Nakanishi, K. M., Fujii, K., & Todo, S. (2020). Sequential minimal optimization for quantum-classical hybrid algorithms. *Physical Review Research*, *2*(4), 043158.
117. Nielsen, M. A., & Chuang, I. (2019). *Quantum computation and quantum information*. Cambridge University Press.
118. Office of Science and Technology Policy. (2023). National quantum initiative. *The White House*.
119. Orús, R., Mugel, S., & Lizaso, E. (2019). Quantum computing for finance: Overview and prospects. *Reviews in Physics*, *4*, 100028.
120. Ostaszewski, M., Mishra, U., & Traore, M. (2021). Quantum optimal control using phase-modulated driving fields. *New Journal of Physics*, *23*(6), 063003.
121. Park, D. K., Petruccione, F., & Rhee, J. K. K. (2020). Circuit-based quantum random access memory for classical data. *Scientific Reports*, *9*(1), 3949.
122. Pepper, A., Tischler, N., & Pryde, G. J. (2019). Experimental realization of a quantum autoencoder: The compression of qutrits via machine learning. *Physical Review Letters*, *122*(6), 060501.
123. Peres, A. (2020). *Quantum theory: Concepts and methods*. Springer.
124. Perdomo-Ortiz, A., Benedetti, M., Realpe-Gómez, J., & Biswas, R. (2019). Opportunities and challenges for quantum-assisted machine learning in near-term quantum computers. *Quantum Science and Technology*, *3*(3), 030502.
125. Peruzzo, A., McClean, J., Shadbolt, P., Yung, M. H., Zhou, X. Q., Love, P. J., Aspuru-Guzik, A., & O'Brien, J. L. (2020). A variational eigenvalue solver on a photonic quantum processor. *Nature Communications*, *5*(1), 4213.
126. Pichler, H., Choi, S., Zoller, P., & Lukin, M. D. (2020). Universal quantum computations with a symmetric subspace code. *Physical Review A*, *95*(5), 052306.
127. Pino, J. M., Dreiling, J. M., Figgatt, C., Gaebler, J. P., Moses, S. A., Allman, M. S., Baldwin, C. H., Foss-Feig, M., Hayes, D., Mayer, K., Ryan-Anderson, C., & Neyenhuis, B. (2021). Demonstration of the trapped-ion quantum CCD computer architecture. *Nature*, *592*(7853), 209-213.
128. Powers, D. M. (2020). Evaluation: From precision, recall and F-measure to ROC, informedness, markedness and correlation. *arXiv preprint arXiv:2010.16061*.
129. Preskill, J. (2021). Quantum computing 40 years later. *arXiv preprint arXiv:2106.10522*.

130. Preskill, J. (2023). Quantum computing in the NISQ era and beyond. *Quantum*, *2*, 79.
131. Proctor, T. J., Rudinger, K., Young, K., Sarovar, M., & Blume-Kohout, R. (2021). What randomized benchmarking actually measures. *Physical Review Letters*, *119*(13), 130502.
132. Qian, L., Sun, J., Zhang, Y., & Gao, J. (2022). Quantum kernel methods for pattern recognition. *IEEE Transactions on Neural Networks and Learning Systems*, *33*(10), 5841-5853.
133. Rajpurkar, P., Irvin, J., Ball, R. L., Zhu, K., Yang, B., Mehta, H., Duan, T., Ding, D., Bagul, A., Langlotz, C. P., Patel, B. N., Yeom, K. W., Shpanskaya, K., Blankenberg, F. G., Seekins, J., Amrhein, T. J., Mong, D. A., Halabi, S. S., Zucker, E. J., ... & Lungren, M. P. (2019). Deep learning for chest radiograph diagnosis: A retrospective comparison of the CheXNeXt algorithm to practicing radiologists. *PLoS Medicine*, *15*(11), e1002686.
134. Rarity, J. (2021). Quantum technology landscape and roadmap in China. *International Journal of Quantum Information*, *18*(02), 1941003.
135. Ristè, D., Da Silva, M. P., Ryan, C. A., Cross, A. W., Córcoles, A. D., Smolin, J. A., Gambetta, J. M., Chow, J. M., & Johnson, B. R. (2021). Demonstration of quantum advantage in machine learning. *npj Quantum Information*, *3*(1), 16.
136. Rol, M. A., Ciorciaro, L., Malinowski, F. K., Tarasinski, B. M., Sagastizabal, R. E., Bultink, C. C., Salathe, Y., Haandbæk, N., Sedivy, J., & DiCarlo, L. (2020). Time-domain characterization and correction of on-chip distortion of control pulses in a quantum processor. *Applied Physics Letters*, *116*(5), 054001.
137. Romero, J., Olson, J. P., & Aspuru-Guzik, A. (2019). Quantum autoencoders for efficient compression of quantum data. *Quantum Science and Technology*, *2*(4), 045001.
138. Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (2019). Learning representations by back-propagating errors. *Cognitive Modeling*, *5*(3), 1.
139. Ryan-Anderson, C., Bohnet, J. G., Lee, K., Gresh, D., Hankin, A., Gaebler, J. P., Francois, D., Chernoguzov, A., Lucchetti, D., Brown, N. C., Gatterman, T. M., Halit, S. K., Gilmore, K., Gerber, J. A., Neyenhuis, B., Hayes, D., & Stutz, R. P. (2021). Realization of real-time fault-tolerant quantum error correction. *Physical Review X*, *11*(4), 041058.
140. Sarovar, M., & Proctor, T. (2020). Detecting crosstalk errors in quantum information processors. *Quantum*, *4*, 321.
141. Sarma, S. D., Freedman, M., & Nayak, C. (2019). Majorana zero modes and topological quantum computation. *npj Quantum Information*, *1*(1), 15001.
142. Sasaki, M., & Carlini, A. (2022). Quantum learning and universal quantum matching machine. *Physical Review A*, *66*(2), 022303.
143. Schölkopf, B., & Smola, A. J. (2020). *Learning with kernels: Support vector machines, regularization, optimization, and beyond*. MIT Press.
144. Schuld, M. (2021). Supervised quantum machine learning models are kernel methods. *arXiv preprint arXiv:2101.11020*.
145. Schuld, M., Bergholm, V., Gogolin, C., Izaac, J., & Killoran, N. (2019). Evaluating analytic gradients on quantum hardware. *Physical Review A*, *99*(3), 032331.
146. Schuld, M., Bocharov, A., Svore, K. M., & Wiebe, N. (2020). Circuit-centric quantum classifiers. *Physical Review A*, *101*(3), 032308.
147. Schuld, M., & Killoran, N. (2019). Quantum machine learning in feature Hilbert spaces. *Physical Review Letters*, *122*(4), 040504.
148. Schuld, M., & Petruccione, F. (2021). *Machine learning with quantum computers*. Springer.
149. Shawe-Taylor, J., & Cristianini, N. (2021). *Kernel methods for pattern analysis*. Cambridge University Press.
150. Shor, P. W. (2019). Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM Review*, *41*(2), 303-332.
151. Sim, S., Johnson, P. D., & Aspuru-Guzik, A. (2019). Expressibility and entangling capability of parameterized quantum circuits for hybrid quantum-classical algorithms. *Advanced Quantum Technologies*, *2*(12), 1900070.

152. Slussarenko, S., & Pryde, G. J. (2019). Photonic quantum information processing: A concise review. *Applied Physics Reviews*, *6*(4), 041303.
153. Sokolova, M., & Lapalme, G. (2019). A systematic analysis of performance measures for classification tasks. *Information Processing & Management*, *45*(4), 427-437.
154. Sousa, J. D., Martins, A. C., Leitão, B., & da Silva, C. (2023). Quantum computational chemistry. *WIREs Computational Molecular Science*, *13*(1), e1627.
155. Spirtes, P. (2020). Quantum computing: Progress and prospects. *National Academies Press*.
156. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2021). Dropout: A simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, *15*(1), 1929-1958.
157. Stehman, S. V. (2021). Selecting and interpreting measures of thematic classification accuracy. *Remote Sensing of Environment*, *62*(1), 77-89.
158. Steinbrecher, G. R., Olson, J. P., Englund, D., & Carolan, J. (2019). Quantum optical neural networks. *npj Quantum Information*, *5*(1), 60.
159. Strikis, A., Qin, D., Chen, Y., Benjamin, S. C., & Li, Y. (2021). Learning-based quantum error mitigation. *PRX Quantum*, *2*(4), 040330.
160. Suzuki, Y., Kawase, Y., Masumura, Y., Hiraga, Y., Nakadai, M., Chen, J., Nakanishi, K. M., Mitarai, K., Imai, R., Tamiya, S., Yamamoto, T., Yan, T., Kawakubo, T., Nakano, Y., Ibe, Y., Zhang, Y., Yamashita, H., Yoshimura, H., Hayashi, A., & Fujii, K. (2020). Qulacs: A fast and versatile quantum circuit simulator for research purpose. *Quantum*, *5*, 559.
161. Svore, K., Geller, A., Troyer, M., Azariah, J., Granade, C., Heim, B., Kliuchnikov, V., Mykhailova, M., Paz, A., & Roetteler, M. (2019). Q#: Enabling scalable quantum computing and development with a high-level DSL. *Proceedings of the Real World Domain Specific Languages Workshop 2018*, 1-10.
162. Sweke, R., Wilde, F., Meyer, J. J., Schuld, M., Fährmann, P. K., Meynard-Piganeau, B., & Eisert, J. (2020). Stochastic gradient descent for hybrid quantum-classical optimization. *Quantum*, *4*, 314.
163. Tannu, S. S., & Qureshi, M. K. (2019). Not all qubits are created equal: A case for variability-aware policies for NISQ-era quantum computers. *Proceedings of the Twenty-Fourth International Conference on Architectural Support for Programming Languages and Operating Systems*, 987-999.
164. Temme, K., Bravyi, S., & Gambetta, J. M. (2023). Error mitigation for short-depth quantum circuits. *Physical Review Letters*, *119*(18), 180509.
165. Terhal, B. M. (2019). Quantum error correction for quantum memories. *Reviews of Modern Physics*, *87*(2), 307.
166. Van Der Maaten, L., Postma, E., & Van den Herik, J. (2019). Dimensionality reduction: A comparative review. *Journal of Machine Learning Research*, *10*(66-71), 13.
167. Vepsäläinen, A. P., Karamlou, A. H., Orrell, J. L., Dogra, A. S., Loer, B., Vasconcelos, F., Kim, D. K., Melville, A. J., Niedzielski, B. M., Yoder, J. L., Gustavsson, S., Formaggio, J. A., VanDevender, B. A., & Oliver, W. D. (2020). Impact of ionizing radiation on superconducting qubit coherence. *Nature*, *584*(7822), 551-556.
168. Walter, T., Kurpiers, P., Gasparinetti, S., Magnard, P., Potočnik, A., Salathé, Y., Pechal, M., Mondal, M., Oppliger, M., Eichler, C., & Wallraff, A. (2019). Rapid high-fidelity single-shot dispersive readout of superconducting qubits. *Physical Review Applied*, *7*(5), 054020.
169. Wang, S., Fontaine, N. K., Liao, S., Cassabois, G., Maring, N., Cao, H., Guo, X., Liu, K., Zhao, Z., Zhang, Q., Pan, J. W., & Wehner, S. (2022). Quantum memory for photonic qubits. *Nature Photonics*, *13*(5), 323-327.
170. Wright, K., Beck, K. M., Debnath, S., Amini, J. M., Nam, Y., Grzesiak, N., Chen, J. S., Pisenti, N. C., Chmielewski, M., Collins, C., Hudek, K. M., Mizrahi, J., Wong-Campos, J. D., Allen, S., Apisdorf, J., Solomon, P., Williams, M., Ducore, A. M., Blinov, A., ... & Monroe, C. (2019). Benchmarking an 11-qubit quantum computer. *Nature Communications*, *10*(1), 5464.
171. Xanadu. (2023). Xanadu quantum cloud. *Xanadu Quantum Technologies Inc*.
172. Xia, R., & Valiant, G. (2019). Training a quantum neural network to solve the

contextual multi-armed bandit problem. *arXiv preprint arXiv:1905.10528*.

173. Yang, C. H., Leon, R. C. C., Hwang, J. C. C., Saraiva, A., Tanttu, T., Huang, W., Camirand Lemyre, J., Chan, K. W., Tan, K. Y., Hudson, F. E., Itoh, K. M., Morello, A., Pioro-Ladrière, M., Laucht, A., & Dzurak, A. S. (2021). Operation of a silicon quantum processor unit cell above one kelvin. *Nature*, *580*(7803), 350-354.

174. Zulehner, A., Paler, A., & Wille, R. (2019). An efficient methodology for mapping quantum circuits to the IBM QX architectures. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, *38*(7), 1226-1236.

Conflict of Interest: No Conflict of Interest
Source of Funding: Author(s) Funded the Research
How to Cite: Raj, A. (2026). Quantum Machine Learning: Bridging Quantum Computing and AI for Next-Generation Pattern Recognition. *Frontiers in Emerging Technology, 2*(1), 01-24