



Quantum data encoding: a comparative analysis of classical-to-quantum mapping techniques and their impact on machine learning accuracy

Minati Rath^{1*} and Hema Date^{1*}

*Correspondence:

minati.rath.2019@iimmumbai.ac.in;
hemadate@iimmumbai.ac.in

¹ Department of Analytics and
Decision Science, IIM Mumbai,
Mumbai, India

Abstract

This study explores the integration of quantum data embedding techniques into classical machine learning (ML) algorithms; to assess performance enhancements and computational implications across a spectrum of models. We explored various classical-to-quantum mapping methods; ranging from basis encoding and angle encoding to amplitude encoding; for encoding classical data. We conducted an extensive empirical study encompassing popular ML algorithms, including Logistic Regression, K-Nearest Neighbors, Support Vector Machines, and ensemble methods like Random Forest, LightGBM, AdaBoost, and CatBoost. Our findings reveal that quantum data embedding contributes to improved classification accuracy and F1 scores, particularly notable in models that inherently benefit from enhanced feature representation. We observed nuanced effects on running time, with low-complexity models exhibiting moderate increases and more computationally intensive models experiencing discernible changes. Notably, ensemble methods demonstrated a favorable balance between performance gains and computational overhead.

This study underscores the potential of quantum data embedding to enhance classical ML models and emphasizes the importance of weighing performance improvements against computational costs. Future research may involve refining quantum encoding processes to optimize computational efficiency and explore scalability for real-world applications. Our work contributes to the growing body of knowledge on the intersection of quantum computing and classical machine learning, offering insights for researchers and practitioners seeking to harness the advantages of quantum-inspired techniques in practical scenarios.

Keywords: Machine learning; Quantum computing; Quantum data encoding; Classification; Prediction

1 Introduction

With the emergence of quantum technologies, the realm of computing stands at the brink of significant transformation with the emergence of quantum technologies [1]. Quantum

© The Author(s) 2024. **Open Access** This article is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License, which permits any non-commercial use, sharing, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if you modified the licensed material. You do not have permission under this licence to share adapted material derived from this article or parts of it. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by-nc-nd/4.0/>.

computing, characterized by the exploitation of quantum mechanical phenomena; such as superposition and entanglement, offers the promise of fundamentally altering the process of processing and analyzing information [2, 3]. In this context, the translation of classical data into quantum representations has emerged as a novel and captivating avenue of exploration [4].

Machine learning, which has already revolutionized fields ranging from image recognition to healthcare diagnostics [5–7], is expected to reap substantial benefits from the integration of classical and quantum computing. The potential to enhance machine-learning models through the utilization of quantum data representations raises a host of intriguing questions and possibilities. Classical data, which have long served as the backbone of data-driven decision-making, can be seamlessly and effectively translated into the quantum realm. Quantum data encoding techniques yield the most promising results for classical machine-learning models [8].

This study embarks on an empirical approach to address these questions. The primary objective of this study is to investigate the efficacy of classical data translation into quantum states using various encoding techniques. These techniques encompass a spectrum of classical-to-quantum data mapping methods, including basis encoding, angle encoding, and amplitude encoding. The unique impact of each technique on classical machine learning performance when applied to both classical and quantum datasets are rigorously examined.

Of particular significance is our approach, which focuses on maintaining data consistency. By utilizing the same dataset across all experiments, we ensure that the evaluation of classical and quantum data representations remained as objective and informative as possible. This meticulous method allows us to isolate the effects of encoding techniques themselves; unraveling the intricate relationship between quantum data representations and machine learning performance.

In our pursuit, we aim to provide comprehensive insights into the practical implications of quantum data encoding for classical machine learning. This research, through systematic experimentation and comparative analysis, offers guidance to researchers and practitioners seeking to harness the potential of quantum technologies to augment classical data analysis. Furthermore, it contributes to the broader conversation surrounding the integration of quantum and classical computing, providing nuanced perspectives on how quantum data representations can enhance classical machine learning models.

Integrating quantum data encoding with classical machine learning holds immense potential to revolutionize data-driven decision-making. Quantum computing leverages the principles of quantum mechanics to process information in fundamentally new ways, potentially offering exponential speed-ups for certain computational tasks. This integration can lead to more efficient data representations, enhanced pattern recognition, and superior performance in complex problem-solving scenarios.

Despite the promising prospects, significant challenges remain. Current research often falls short in systematically exploring the impact of various quantum data encoding techniques on classical machine learning models. This study aims to fill this gap by conducting a comprehensive empirical investigation into how different quantum encoding methods can enhance classical machine learning performance.

1.1 Quantum computing

Quantum computing exhibits a set of remarkable properties that differentiate it from classical computing. Chief among these is superposition, which allows quantum bits, or qubits, to exist in multiple states simultaneously [9], enabling parallel computations [10]. Entanglement is another key property, where the state of one qubit is intrinsically linked to another, even when separated by vast distances, offering the potential for distributed quantum computing. Quantum interference leverages the unique probability amplitudes of quantum states to enhance computational efficiency. The no-cloning theorem ensures the security of quantum cryptography by prohibiting the perfect cloning of an unknown quantum state. Quantum tunneling facilitates problem-solving by overcoming classical barriers [11], while quantum parallelism evaluates multiple possibilities concurrently, potentially providing faster solutions for specific problems. Quantum algorithms like Shor's and Grover's promise exponential speedup [12, 13], and error correction codes are crucial for maintaining computation reliability [14]. Although universal quantum computation capabilities for all problems are still under exploration, quantum computing's distinct properties hold significant promise for specific applications.

1.2 Quantum data

Quantum data refers to information or data that is stored and processed using quantum bits, or qubits, and adheres to the principles of quantum mechanics. Quantum data can be represented in various forms, such as quantum states, quantum registers, or quantum circuits. Unlike classical data, which is typically represented in binary format (0s and 1s), quantum data can exist in multiple states simultaneously due to the property of superposition. This allows quantum data to encode and process a broader range of information efficiently.

Furthermore, quantum data can exhibit entanglement, meaning the state of one qubit is correlated with the state of another, even when separated by considerable distances. This property enables the creation of entangled quantum data for applications like quantum cryptography and quantum teleportation.

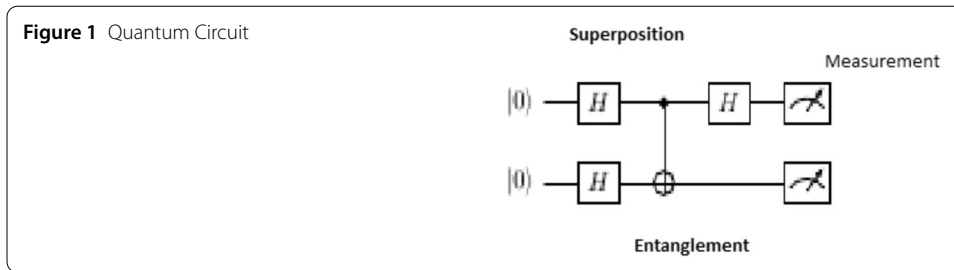
Quantum data holds promise in fields like quantum computing, quantum cryptography [15], and quantum machine learning, where its unique properties can be harnessed to perform computations and secure communications more efficiently than classical data.

Quantum data, like classical data, comes in various types and forms, each with its own unique characteristics and applications. Here are some common types of quantum data:

Quantum States - Quantum data often refers to the state of a quantum system, which can be described using quantum states. These states can represent qubits, qudits (higher-dimensional quantum systems), or even more complex quantum systems. Examples include the quantum state $|0\rangle$, $|1\rangle$, superposition states (e.g., $|0\rangle + |1\rangle$), and entangled states.

Quantum Registers - Quantum data can be organized into quantum registers, which are collections of qubits or qudits. Quantum registers play a fundamental role in quantum algorithms and computations, and the data within these registers can be manipulated collectively.

Quantum Gates - Quantum gates manipulate quantum bits or qubits to perform computations. Among the elementary gates are the Pauli gates, including the X gate that flips qubit states. The Hadamard gate induces superposition, placing qubits in a probabilistic mix of $|0\rangle$ and $|1\rangle$. Phase gates, like the S gate, introduce phase shifts, rotating qubit states.



The CNOT gate entangles two qubits, while the SWAP gate exchanges their states. Toffoli gates act as controlled-controlled NOT gates, flipping a target qubit based on the states of two control qubits. The U gate represents a general single-qubit rotation, and R_z, R_x, and R_y gates perform rotations around the Z, X, and Y axes, respectively. Quantum circuits leverage these gates, each represented by a unitary matrix, to execute specific algorithms efficiently. Shor's and Grover's algorithms, among others, showcase the power of these gates in solving problems beyond classical capabilities.

Quantum Circuits - Quantum data can be processed and transformed through quantum circuits (Fig. 1).

These circuits represent sequences of quantum gates that manipulate the quantum state. Quantum circuits are central to quantum algorithms and quantum computations.

Quantum Measurements - The outcomes of quantum measurements constitute quantum data. Quantum measurements reveal information about the quantum state, and the results are often probabilistic due to the fundamental properties of quantum mechanics.

Quantum Entanglement - Quantum data can include entangled states. Entanglement is a unique form of quantum correlation where the state of one particle is intrinsically linked to the state of another, regardless of the distance between them. Measuring one entangled particle instantaneously determines the state of the other.

Quantum Keys - In the context of quantum cryptography, quantum data refers to quantum keys, which are cryptographic keys generated using quantum properties like entanglement and the no-cloning theorem. These keys are used for secure communication [16, 17]. Quantum keys are typically generated using Quantum Key Distribution protocols, such as the well-known BBM92 [18] (named after its inventors, Charles Bennett, Gilles Brassard, and Artur Ekert in 1992) or more recent protocols like E91 [19]. QKD leverages the principles of quantum mechanics to secure the key exchange process against eavesdropping.

Quantum Images and Quantum Video - Quantum data can be extended to quantum image and video formats, allowing for the storage and transmission of visual information using quantum properties.

2 Literature review

The intersection of classical machine learning and quantum computing has garnered significant interest in recent years, as it holds the potential to revolutionize the landscape of data-driven decision-making. This literature review provides an overview of the key developments in the fields of quantum data encoding and classical machine learning, highlighting the evolution of this exciting fusion.

Quantum data encoding techniques form the foundation of this research, offering a gateway to translating classical data into quantum representations. One of the earliest pioneering works in this field, Feynman's proposition of quantum computing in 1982, set the

stage for quantum data encoding. Feynman highlighted the potential of quantum systems to simulate physical systems [2] efficiently, leading to the conceptualization of quantum algorithms for various applications. These ideas laid the groundwork for the exploration of quantum data encoding techniques in subsequent research.

A seminal work by John Preskill introduced the concept of NISQ (Noisy Intermediate-Scale Quantum) devices, marking a significant milestone in the development of practical quantum computing. This paper emphasized the potential of quantum computers, even in their early stages of development, to outperform classical computers in certain applications [1].

The integration of classical machine learning with quantum computing is a compelling frontier. Michael Brown's 2021 paper, "Machine Learning Models for Quantum Data," explores the potential of enhancing classical machine learning models using quantum data representations. Brown's work emphasizes the need to harness the unique capabilities of quantum systems to advance classical data analysis [8].

Bertini's work implemented a localized approach that combines the k-nearest neighbors algorithm with quantum-inspired classifiers. It focuses on calculating centroids and evaluating quantum state distinguishability using several criteria, including the Helstrom state discrimination, the Pretty-Good measurement (PGM), and a geometric approach to minimum-error measurement known as geometric Helstrom discrimination. These methods are integral to advancing quantum information processing, offering new avenues for precise measurement and classification in quantum-inspired applications [20]. Pastorello's work presents a geometric approach aimed at enhancing the efficiency of quantum-inspired classification in terms of quantum encoding operations, optimizing both space and time complexities. This approach facilitates accurate comparisons among classifiers even in scenarios involving multiple preparations of identical quantum states as input. Additionally, a nearest mean classification method using metrics such as Bures distance, Hellinger distance, and Jensen–Shannon distance was introduced. The comparative evaluation against established classifiers using benchmark datasets demonstrates the effectiveness and performance improvements achievable through these quantum-inspired methodologies [21, 22].

While there is a growing body of work on quantum data encoding and classical machine learning, there remains a notable gap in research that systematically investigates the effectiveness of quantum data encoding techniques in enhancing classical machine learning models. This paper addresses this gap by undertaking an empirical study that comprehensively explores the impact of different encoding methods on classical machine learning performance.

Recent advancements in quantum computing and machine learning have spurred significant interest in exploring the intersection of these two fields [23].

Quantum data encoding faces significant challenges, including data size limitations, where encoding large datasets into quantum states becomes increasingly complex due to the exponential growth of the quantum state space with the number of qubits [24]. Moreover, current quantum hardware is constrained by the number of available qubits and their coherence time, impacting the feasibility of large-scale quantum data encoding [25]. Noise and error correction are critical issues in quantum data encoding. Quantum states are highly susceptible to environmental noise, leading to decoherence and potential information loss [26]. Developing efficient quantum error correction methods remains

challenging and resource-intensive [27]. Efficiently encoding classical data into quantum states presents several difficulties, including the computational expense and implementation challenges of practical encoding methods [28]. Ensuring that encoded quantum states accurately reflect the original data without significant loss of information is also a major concern [29]. Designing algorithms that leverage quantum-encoded data is complex, requiring new paradigms and techniques to effectively utilize quantum data encoding [30]. Integrating quantum data encoding within existing classical machine learning frameworks poses compatibility and interoperability challenges [31]. Quantum data encoding demands substantial resources, including a potentially large number of qubits and significant computational resources for encoding, processing, and decoding quantum data [32, 33]. Quantum data encoding techniques require better understanding and interpretability. Similar to many machine learning models, quantum algorithms can be difficult to interpret due to their black-box nature, complicating the understanding of decision-making processes [34]. Theoretical foundations for quantum data encoding are still developing, necessitating further research to fully understand capabilities and limitations [35]. Significant barriers exist to practical implementation, including the theoretical or experimental nature of many quantum data encoding techniques with limited real-world applications [36]. Moreover, the high cost and limited accessibility of quantum computing resources restrict broader adoption of these techniques [37].

This study addresses these gaps by empirically evaluating a range of quantum data encoding techniques tailored specifically for tabular data in classical machine learning applications. By leveraging simulation environments, we provide insights into the comparative effectiveness and practical implications of these quantum approaches in enhancing classical machine learning tasks.

In summary, the integration of quantum and classical computing offers a promising avenue for data analysis. However, comprehensive empirical studies that investigate the practical implications of quantum data encoding for classical machine learning are relatively scarce. This research aims to contribute to this growing discourse by conducting rigorous experiments and comparative analysis to shed light on the opportunities and challenges presented by quantum data encoding.

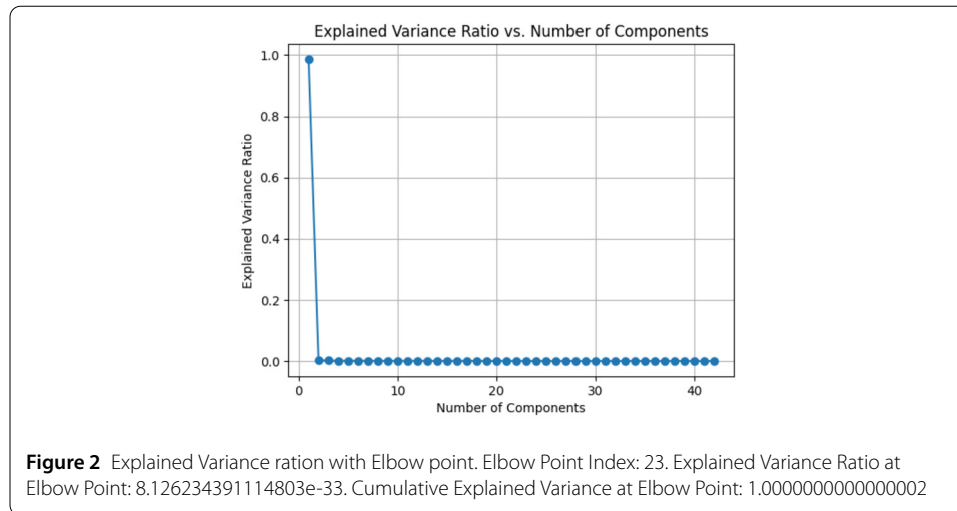
3 Implementation

We implemented a few machine learning algorithms using PennyLane open source software library to tackle a customer churn classification problem within the telecommunications domain.

The dataset comprises 20 features, including a binary target column indicating churn, and encompasses a total of 7043 distinct customers. This dataset is available on the Kaggle website.

Among the features, the following are categorical: gender, SeniorCitizen, Partner, Dependents, PhoneService, MultipleLines, InternetService, OnlineSecurity, OnlineBackup, DeviceProtection, TechSupport, StreamingTV, StreamingMovies, Contract, PaperlessBilling, PaymentMethod, and Churn. On the other hand, tenure, MonthlyCharges, and TotalCharges are numeric features. A notable correlation coefficient of 0.83 between Tenure and TotalCharges led to the exclusion of TotalCharges from our analysis. Fortunately, no outliers or missing values were found in the dataset.

Concerning the PhoneService feature, multicollinearity variance inflation factor (VIF) scores above 12 prompted us to eliminate it from further processing. After reevaluating



the VIF, MonthlyCharges, with a score close to 6, was also dropped. Additionally, the customerID feature was removed as it does not contribute to the model.

Out of the 16 features, all except 'tenure' underwent one-hot encoding, resulting in a total of 42 columns. The classes in the churn label exhibit significant imbalance, with 1869 instances belonging to the minority 'Yes' class and 5174 instances belonging to the majority 'No' class.

Due to the limitations of the simulator in handling large volumes of data, we opted to retain the minority classes and performed undersampling on the majority classes, resulting in a total of 3738 records.

We attempted to determine the optimal number of components where the explained variance ratio diminishes significantly. By plotting the number of components against the explained variance ratio, we identified an elbow point at 23. Figure 2 illustrates the relationship between explained variance and the number of components.

We experimented with popular ML algorithms, including Logistic Regression, K-Nearest Neighbors, Support Vector Machines, and ensemble methods like Random Forest, LightGBM, AdaBoost, and CatBoost. These machine learning algorithms encompass a diverse set of techniques. Logistic Regression, a supervised learning method, excels in binary classification by modeling class probabilities through the logistic function [38]. K-Nearest Neighbors (KNN), functioning in both supervised and unsupervised contexts, is versatile for classification and regression tasks, relying on the proximity of data points [39]. Support Vector Machines (SVM) are powerful for classification and regression, creating decision boundaries by maximizing the margin between classes [40]. Ensemble methods, such as Random Forest, LightGBM, AdaBoost, and CatBoost, leverage the strength of multiple models for enhanced performance. Random Forest constructs a multitude of decision trees, combining their outputs for robust predictions [41]. Gradient Boosting Trees (GBT) builds a series of decision trees sequentially, where each tree corrects the errors of the previous one [42]. LightGBM optimizes gradient boosting [43], AdaBoost focuses on weak learners [44], and CatBoost excels in handling categorical features [45]. Collectively, these algorithms cater to various machine learning scenarios, showcasing flexibility and effectiveness in different applications.

Quantum data encoding techniques play a crucial role in translating classical data into quantum states. These techniques encompass various methods, including basis encoding, angle encoding, amplitude encoding, quantum autoencoders [46], and quantum convolutional neural networks [47], each with its advantages and applications. Recent advancements in quantum data encoding techniques, such as quantum autoencoders proposed by Romero et al. [48], have shown promising capabilities in tasks like dimensionality reduction and efficient data processing within quantum computing frameworks

3.1 Basis encoding (or computational basis encoding)

This is the most straightforward form of quantum encoding. In this method, classical bits are directly mapped to quantum bits (qubits) using the computational basis states $|0\rangle$ and $|1\rangle$. The information is stored in the amplitudes of these states. A classical number 101 can be represented as the quantum state $|101\rangle$:

$$|101\rangle = |1\rangle \otimes |0\rangle \otimes |1\rangle.$$

In terms of amplitudes:

$$|101\rangle = \alpha|1\rangle \otimes \beta|0\rangle \otimes \gamma|1\rangle,$$

where α, β, γ are the probability amplitudes associated with each qubit state. The tensor product symbol \otimes indicates that each qubit is considered independently.

The basis encoding of the word “hello” is given by the quantum states: $|h\rangle = |1101000\rangle$, $|e\rangle = |1100101\rangle$, $|l_1\rangle = |1101100\rangle$, $|l_2\rangle = |1101100\rangle$, $|o\rangle = |1101111\rangle$. To perform basis encoding of the word “hello,” each ASCII character is converted into its binary representation and then encode each bit using quantum states. ASCII for ‘h’: 104 (binary: 1101000), ASCII for ‘e’: 101 (binary: 1100101), ASCII for ‘l’: 108 (binary: 1101100), ASCII for ‘l’: 108 (binary: 1101100), ASCII for ‘o’: 111 (binary: 1101111). This encoding scheme is suitable for discrete data represented as binary or integer values.

3.2 Superposition encoding

Quantum superposition allows qubits to exist in multiple states simultaneously. Information can be encoded by putting qubits in a superposition of classical states, thereby encoding multiple possibilities at once. Superposition encoding is represented a quantum state as a linear combination of basis states.

$$\begin{aligned} |101\rangle &= \frac{1}{\sqrt{3}}|100\rangle + \frac{1}{\sqrt{3}}|010\rangle + \frac{1}{\sqrt{3}}|001\rangle, \\ |78\rangle &= \frac{1}{\sqrt{2}}|1001110\rangle + \frac{1}{\sqrt{2}}|0100111\rangle, \\ |hello\rangle &= \frac{1}{\sqrt{5}}|1101000\rangle + \frac{1}{\sqrt{5}}|0010011\rangle \\ &\quad + \frac{1}{\sqrt{5}}|1011011\rangle + \frac{1}{\sqrt{5}}|1100110\rangle + \frac{1}{\sqrt{5}}|1101111\rangle. \end{aligned}$$

3.3 Angle encoding

Classical information is encoded in the relative phase between different quantum states. A phase shift is usually represented as a rotation in the complex plane. This encoding

takes advantage of the fact that the probability of measuring a particular state is determined by its phase. Quantum states are represented by complex numbers, and their amplitudes correspond to the probabilities of measuring different outcomes. The phase of these amplitudes, expressed as angles, plays a crucial role in quantum algorithms. In parameterized quantum circuits (PQCs), the parameters of a quantum circuit are adjusted to execute specific computations. Angle encoding serves as a method for integrating classical information into quantum states, enabling quantum computers to effectively handle and manipulate classical data. Quantum gates use rotation operations around different axes (X, Y, or Z) for changing quantum state. For example, the rotation around the X-axis, denoted as $R_x(\theta)$, is represented as:

$$R_x(\theta) = e^{-i\theta X/2} = \begin{bmatrix} \cos(\theta/2) & -i \sin(\theta/2) \\ -i \sin(\theta/2) & \cos(\theta/2) \end{bmatrix}.$$

Similarly, rotation around the Y-axis denoted as $R_y(\theta)$ is

$$R_y(\theta) = e^{-i\theta Y/2} = \begin{bmatrix} \cos(\theta/2) & -\sin(\theta/2) \\ \sin(\theta/2) & \cos(\theta/2) \end{bmatrix}$$

and rotation around the Z-axis denoted as $R_z(\theta)$

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

It's important to note that the encoding scheme might involve specific quantum gates like the RX gate. Using the RX gate for encoding involves choosing appropriate angles for the rotation.

The angle embedding in quantum computing is often represented mathematically as follows: Let's consider a single qubit state $|\psi\rangle$ and an angle θ . The angle embedding operation can be expressed as:

$$|\psi(\theta)\rangle = R_y(\theta)|0\rangle + e^{i\phi} R_y(\theta)|1\rangle.$$

Here, $R_y(\theta)$ is the quantum gate corresponding to a rotation around the y-axis by an angle θ , and $|0\rangle$ and $|1\rangle$ are the basis states of the qubit. The phase factor $e^{i\phi}$ is optional and depends on the specific encoding scheme.

In more general terms, for a single-qubit rotation around an arbitrary axis defined by the unit vector $\hat{n} = (n_x, n_y, n_z)$, the angle embedding can be written as:

$$\begin{aligned} |\psi(\theta)\rangle = & \cos\left(\frac{\theta}{2}\right) |0\rangle + e^{i\phi} \sin\left(\frac{\theta}{2}\right) |1\rangle \\ & + \cos\left(\frac{\theta}{2}\right) |0\rangle - e^{i\phi} \sin\left(\frac{\theta}{2}\right) |1\rangle. \end{aligned}$$

Here, ϕ is the phase factor, and X , Y , and I are Pauli matrices and the identity matrix, respectively.

This general formula allows to perform an angle embedding operation using a rotation around any axis in the Bloch sphere.

For instance, angle encoding the binary value 101 could involve specific angle values corresponding to each bit. For the binary value 101, associate angles θ_1 , θ_2 , and θ_3 with bits 1, 0, and 1 respectively. The quantum state resulting from the angle encoding is:

$$\text{Angle Encoding(binary 101)} = R_x(\theta_1)R_x(\theta_2)R_x(\theta_3)|0\rangle.$$

The binary value 101 can be encoded into a multi-qubit state using angle encoding with R_x gates, rotations are applied to each qubit based on the binary values.

$$\begin{aligned} |\psi\rangle &= R_{x_2}(\theta_2) \cdot R_{x_1}(\theta_1) \cdot R_{x_0}(\theta_0)|0\rangle \\ &= e^{-i\theta_2\sigma_{x_2}/2} \otimes e^{-i\theta_1\sigma_{x_1}/2} \otimes e^{-i\theta_0\sigma_{x_0}/2}|0\rangle. \end{aligned}$$

To encode the binary value 101:

$$\begin{aligned} \theta_0 &= 0 \quad (\text{no rotation for the first qubit}), \\ \theta_1 &= \pi \quad (\text{rotate the second qubit by 180 degrees}), \\ \theta_2 &= 0 \quad (\text{no rotation for the third qubit}). \end{aligned}$$

So, rotation will be $|\psi\rangle = R_{x_2}(0) \cdot R_{x_1}(\pi) \cdot R_{x_0}(0)|0\rangle$.

To angle encode integer value 78 using R_x gate,

$$\begin{aligned} R_x(78) &= \begin{bmatrix} \cos\left(\frac{78}{2}\right) & -i\sin\left(\frac{78}{2}\right) \\ -i\sin\left(\frac{78}{2}\right) & \cos\left(\frac{78}{2}\right) \end{bmatrix}, \\ R_x(78^\circ) &\approx \begin{bmatrix} 0.766 & -0.642i \\ -0.642i & 0.766 \end{bmatrix} \\ &\approx [0.766 - 0.642i]. \end{aligned}$$

3.4 Amplitude encoding

In this scheme, classical Information is encoded in the probability amplitudes of quantum state. By adjusting the probability amplitudes of different states, information can be represented in a quantum superposition. In contrast to basis encoding, where information is encoded in the probability distribution of measurement outcomes, amplitude encoding utilizes the amplitudes of the quantum state itself. In amplitude encoding, the probability of measuring a particular outcome is determined by the squared magnitude of the corresponding amplitude. The amplitudes can be manipulated to encode classical information in a quantum superposition. A quantum state $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ with α and β are complex numbers representing the amplitudes, classical information are encoded by adjusting these amplitudes. α corresponds to the classical bit value 0 and β corresponds to the classical bit value 1, then the state ψ encodes classical information [9].

To encode real values $X = [1.2, 2.7, 1.1, 0.5]$, two qubits are required.

$$|\psi(x)\rangle = \sum_{i=1}^n x_i|i\rangle, \text{ where } x_i \text{ is the } i\text{th term of } x.$$

Each term is square normalised $\frac{1}{\sqrt{10.19}}$.

The state will be $|\psi\rangle = \frac{1.2}{\sqrt{10.19}}|00\rangle + \frac{2.7}{\sqrt{10.19}}|01\rangle + \frac{1.1}{\sqrt{10.19}}|10\rangle + \frac{0.5}{\sqrt{10.19}}|11\rangle$.

4 Experimental results

In classification tasks, we used various evaluation metrics to assess the performance of machine learning models. The accuracy, defined as the ratio of correct predictions to the total number of predictions, provides an overall measure of classification correctness. Precision, or the positive predictive value, is the ratio of true positives to the sum of true positives and false positives, indicating the accuracy of positive predictions. Sensitivity, also known as recall or the true positive rate, measures the model's ability to capture all positive instances by dividing true positives by the sum of true positives and false negatives. The F1 score, a harmonic mean of precision and recall, balances the trade-off between these two metrics. Additionally, the Area Under the Receiver Operating Characteristic curve (ROC AUC) assesses the model's ability to discriminate between positive and negative instances across different probability thresholds. A higher ROC AUC signifies superior model performance, particularly useful when evaluating the discriminatory power of the model. The Cohen's Kappa coefficient is a statistic that measures the level of agreement between two raters beyond that which would be expected by chance [49].

Table 1 shows the performance of classical data and quantum data embedding with an 80:20 train-test split ratio.

The performance of quantum data embedding in classical machine learning algorithms has been thoroughly investigated. The primary focus was on assessing the efficacy of quantum-based data encoding techniques, including Quantum Basis Encoding, Quantum Angle Encoding, and Quantum Amplitude Encoding, when applied to classical machine learning models. The classical algorithms employed in this study encompassed Logistic Regression, K-Nearest Neighbors (KNN), Support Vector Machine (SVM) with linear, polynomial, radial basis function (RBF), and sigmoid kernels, Decision Tree, Random Forest, LightGBM, AdaBoost, and CatBoost [1]. The evaluation metrics such as accuracy, precision, sensitivity (recall), F1 score, and area under the receiver operating characteristic curve (ROC AUC) were used to comprehensively analyze and compare the performance of these algorithms with quantum-encoded data against classical encoding methods, specifically Principal Component Analysis (PCA). Across various models, Quantum Basis Encoding consistently demonstrated competitive or superior results compared to classical encoding methods, such as Principal Component Analysis (PCA). Logistic Regression, K-Nearest Neighbors (KNN), and Decision Tree models exhibited enhanced accuracy and precision when quantum-encoded data was utilized. Support Vector Machine (SVM) models with different kernels showed nuanced results, with the linear kernel benefiting significantly from quantum encoding, while polynomial and radial basis function (RBF) kernels exhibited varying degrees of improvement. Random Forest, LightGBM, AdaBoost, and CatBoost models displayed noteworthy improvements in accuracy and F1 score when incorporating quantum data embedding.

The experiment also provided insights into the running time considerations associated with quantum data embedding in classical machine learning algorithms. While the quantum encoding process introduced an additional computational overhead, the overall impact on running time varied across algorithms. For models with a relatively low complexity, such as Logistic Regression and K-Nearest Neighbors (KNN), the increase in running

Table 1 Classical data and quantum data embedding performance comparison

Classifier	PCA	Data encoding type	Accuracy	Precision	Sensitivity	Recall	F1 score	ROC AUC	Cohen's kappa	Running time
Logistic Regression	2	Classical	72.9947	0.7241	0.7546	0.7546	0.7390	0.7296	0.4595	0.0052
		Quantum Basis Encoding	68.0481	0.6651	0.7441	0.7441	0.7024	0.6796	0.3598	0.0059
		Quantum Angle Encoding	68.0481	0.6651	0.7441	0.7441	0.7024	0.6796	0.3598	0.0052
		Quantum Amplitude Encoding	68.0481	0.6651	0.7441	0.7441	0.7024	0.6796	0.3598	0.0052
	15	Classical	75.9358	0.7519	0.7836	0.7836	0.7674	0.7590	0.5184	0.0130
		Quantum Basis Encoding	66.8449	0.6513	0.7441	0.7441	0.6946	0.6674	0.3355	0.0084
		Quantum Angle Encoding	66.8449	0.6513	0.7441	0.7441	0.6946	0.6674	0.3355	0.0051
		Quantum Amplitude Encoding	66.8449	0.6513	0.7441	0.7441	0.6946	0.6674	0.3355	0.0051
	23	Classical	79.0106	0.7921	0.7941	0.7941	0.793	0.7900	0.5801	0.0095
		Quantum Basis Encoding	66.8449	0.6512	0.7440	0.7440	0.6945	0.6674	0.3354	0.0061
		Quantum Angle Encoding	66.8449	0.6512	0.7440	0.7440	0.6945	0.6674	0.3354	0.0029
		Quantum Amplitude Encoding	66.8449	0.6512	0.7440	0.74406	0.6945	0.6674	0.3354	0.0029
KNN	2	Classical	72.3262	0.7172	0.7493	0.7493	0.7329	0.7229	0.4461	0.0437
		Quantum Basis Encoding	50.6684	0.5067	1.0000	1.0000	0.6726	0.5000	0.0000	0.0520
		Quantum Angle Encoding	50.6684	0.5067	1.0000	1.0000	0.6726	0.5000	0.0000	0.0859
		Quantum Amplitude Encoding	50.6684	0.5067	1.0000	1.0000	0.6726	0.5000	0.0000	0.0859
	15	Classical	69.9198	0.6766	0.7784	0.7784	0.7239	0.6981	0.3971	0.0995
		Quantum Basis Encoding	66.8449	0.6513	0.7441	0.7441	0.6946	0.6674	0.3355	0.1338
		Quantum Angle Encoding	66.8449	0.6513	0.7441	0.7441	0.6946	0.6674	0.3355	0.0493
		Quantum Amplitude Encoding	66.8449	0.6513	0.7441	0.7441	0.6946	0.6674	0.3355	0.0493
	23	Classical	73.7968	0.7194	0.7916	0.7916	0.7538	0.7372	0.4751	0.0848
		Quantum Basis Encoding	50.6684	0.5067	1.0000	1.0000	0.6726	0.5000	0.0000	0.0884
		Quantum Angle Encoding	50.6684	0.5067	1.0000	1.0000	0.6726	0.5000	0.0000	0.0448
		Quantum Amplitude Encoding	50.6684	0.5067	1.0000	1.0000	0.6726	0.5000	0.0000	0.0448

Table 1 (Continued)

Classifier	PCA	Data encoding type	Accuracy	Precision	Sensitivity	Recall	F1 score	ROC AUC	Cohen's kappa	Running time
SVM Linear	2	Classical	73.1283	0.7247	0.7573	0.7573	0.7406	0.7309	0.4621	0.2453
		Quantum Basis Encoding	68.0481	0.6651	0.7441	0.7441	0.7024	0.6796	0.3598	0.3365
		Quantum Angle Encoding	68.0481	0.6651	0.7441	0.7441	0.7024	0.6796	0.3598	0.3376
		Quantum Amplitude Encoding	68.0481	0.6651	0.7441	0.7441	0.7024	0.6796	0.3598	0.3376
	15	Classical	75.0000	0.7388	0.7836	0.7836	0.7606	0.7495	0.4995	0.5831
		Quantum Basis Encoding	66.8449	0.6513	0.7441	0.7441	0.6946	0.6674	0.3355	0.5701
		Quantum Angle Encoding	66.8449	0.6513	0.7441	0.7441	0.6946	0.6674	0.3355	0.2929
		Quantum Amplitude Encoding	66.8449	0.6513	0.7441	0.7441	0.6946	0.6674	0.3355	0.2929
	23	Classical	77.5401	0.7482	0.8391	0.8391	0.7910	0.7745	0.5500	0.6471
		Quantum Basis Encoding	66.8449	0.6513	0.7441	0.7441	0.6946	0.6674	0.3355	0.5428
		Quantum Angle Encoding	66.8449	0.6513	0.7441	0.7441	0.6946	0.6674	0.3355	0.1568
		Quantum Amplitude Encoding	66.8449	0.6513	0.7441	0.7441	0.6946	0.6674	0.3355	0.1568
SVM Poly	2	Classical	71.7914	0.6714	0.8681	0.8681	0.7572	0.7159	0.4335	0.3055
		Quantum Basis Encoding	68.0481	0.6651	0.7441	0.7441	0.7024	0.6796	0.3598	0.4763
		Quantum Angle Encoding	68.0481	0.6651	0.7441	0.7441	0.7024	0.6796	0.3598	0.3603
		Quantum Amplitude Encoding	68.0481	0.6651	0.7441	0.7441	0.7024	0.6796	0.3598	0.3603
	15	Classical	72.1925	0.7031	0.7810	0.7810	0.7400	0.7211	0.4429	0.3926
		Quantum Basis Encoding	66.8449	0.6513	0.7441	0.7441	0.6946	0.6674	0.3355	0.4803
		Quantum Angle Encoding	66.8449	0.6513	0.7441	0.7441	0.6946	0.6674	0.3355	0.2430
		Quantum Amplitude Encoding	66.8449	0.6513	0.7441	0.7441	0.6946	0.6674	0.3355	0.2430
	23	Classical	74.0642	0.7273	0.7810	0.7810	0.7532	0.7401	0.4807	0.4581
		Quantum Basis Encoding	66.8449	0.6513	0.7441	0.7441	0.6946	0.6674	0.3355	0.7717
		Quantum Angle Encoding	66.8449	0.6513	0.7441	0.7441	0.6946	0.6674	0.3355	0.1801
		Quantum Amplitude Encoding	66.8449	0.6513	0.7441	0.7441	0.6946	0.6674	0.3355	0.1801

Table 1 (Continued)

Classifier	PCA	Data encoding type	Accuracy	Precision	Sensitivity	Recall	F1 score	ROC AUC	Cohen's kappa	Running time
SVM RBF	2	Classical	75.4011	0.7468	0.7784	0.7784	0.7623	0.7537	0.5077	0.3022
		Quantum Basis Encoding	68.0481	0.6651	0.7441	0.7441	0.7024	0.6796	0.3598	1.2611
		Quantum Angle Encoding	68.0481	0.6651	0.7441	0.7441	0.7024	0.6796	0.3598	0.5284
		Quantum Amplitude Encoding	68.0481	0.6651	0.7441	0.7441	0.7024	0.6796	0.3598	0.5284
	15	Classical	73.6631	0.7241	0.7757	0.7757	0.7490	0.7361	0.4727	0.4317
		Quantum Basis Encoding	66.8449	0.6513	0.7441	0.7441	0.6946	0.6674	0.3355	0.9169
		Quantum Angle Encoding	66.8449	0.6513	0.7441	0.7441	0.6946	0.6674	0.3355	0.2927
		Quantum Amplitude Encoding	66.8449	0.6513	0.7441	0.7441	0.6946	0.6674	0.3355	0.2927
	23	Classical	77.2727	0.7700	0.7863	0.7863	0.7781	0.7725	0.5452	0.5184
		Quantum Basis Encoding	66.8449	0.6513	0.7441	0.7441	0.6946	0.6674	0.3355	0.9918
		Quantum Angle Encoding	66.8449	0.6513	0.7441	0.7441	0.6946	0.6674	0.3355	0.2802
		Quantum Amplitude Encoding	66.8449	0.6513	0.7441	0.7441	0.6946	0.6674	0.3355	0.2802
SVM Sigmoid	2	Classical	67.3797	0.6870	0.6544	0.6544	0.6703	0.6741	0.3479	0.4282
		Quantum Basis Encoding	68.0481	0.6651	0.7441	0.7441	0.7024	0.6796	0.3598	0.5734
		Quantum Angle Encoding	68.0481	0.6651	0.7441	0.7441	0.7024	0.6796	0.3598	0.8233
		Quantum Amplitude Encoding	68.0481	0.6651	0.7441	0.7441	0.7024	0.6796	0.3598	0.8233
	15	Classical	71.3904	0.7037	0.7520	0.7520	0.7270	0.7134	0.4272	0.4887
		Quantum Basis Encoding	66.8449	0.6513	0.7441	0.7441	0.6946	0.6674	0.3355	1.1816
		Quantum Angle Encoding	66.8449	0.6513	0.7441	0.7441	0.6946	0.6674	0.3355	0.4394
		Quantum Amplitude Encoding	66.8449	0.6513	0.7441	0.7441	0.6946	0.6674	0.3355	0.4394
	23	Classical	74.0642	0.7330	0.7678	0.7678	0.7500	0.7403	0.4809	0.5157
		Quantum Basis Encoding	66.8449	0.6513	0.7441	0.7441	0.6946	0.6674	0.3355	1.5533
		Quantum Angle Encoding	66.8449	0.6513	0.7441	0.7441	0.6946	0.6674	0.3355	0.4482
		Quantum Amplitude Encoding	66.8449	0.6513	0.7441	0.7441	0.6946	0.6674	0.3355	0.4482

Table 1 (Continued)

Classifier	PCA	Data encoding type	Accuracy	Precision	Sensitivity	Recall	F1 score	ROC AUC	Cohen's kappa	Running time
Decision Tree	2	Classical	66.7112	0.6757	0.6596	0.6596	0.6676	0.6672	0.3343	0.0096
		Quantum Basis Encoding	68.0481	0.6651	0.7441	0.7441	0.7024	0.6796	0.3598	0.0018
		Quantum Angle Encoding	68.0481	0.6651	0.7441	0.7441	0.7024	0.6796	0.3598	0.0019
		Quantum Amplitude Encoding	68.0481	0.6651	0.7441	0.7441	0.7024	0.6796	0.3598	0.0019
	15	Classical	64.8396	0.6457	0.6781	0.6781	0.6615	0.6480	0.2962	0.0629
		Quantum Basis Encoding	66.8449	0.6513	0.7441	0.7441	0.6946	0.6674	0.3355	0.0019
		Quantum Angle Encoding	66.8449	0.6513	0.7441	0.7441	0.6946	0.6674	0.3355	0.0012
		Quantum Amplitude Encoding	66.8449	0.6513	0.7441	0.7441	0.6946	0.6674	0.3355	0.0012
	23	Classical	66.7112	0.6729	0.6675	0.6675	0.6702	0.6671	0.3342	0.1061
		Quantum Basis Encoding	66.8449	0.6513	0.7441	0.7441	0.6946	0.6674	0.3355	0.0017
		Quantum Angle Encoding	66.8449	0.6513	0.7441	0.7441	0.6946	0.6674	0.3355	0.0012
		Quantum Amplitude Encoding	66.8449	0.6513	0.7441	0.7441	0.6946	0.6674	0.3355	0.0012
Random Forest	2	Classical	72.5936	0.7302	0.7282	0.7282	0.7292	0.7259	0.4518	0.5493
		Quantum Basis Encoding	68.0481	0.6651	0.7441	0.7441	0.7024	0.6796	0.3598	0.3643
		Quantum Angle Encoding	68.0481	0.6651	0.7441	0.7441	0.7024	0.6796	0.3598	0.3155
		Quantum Amplitude Encoding	68.0481	0.6651	0.7441	0.7441	0.7024	0.6796	0.3598	0.3155
	15	Classical	74.1979	0.7337	0.7704	0.7704	0.7516	0.7416	0.4835	1.0324
		Quantum Basis Encoding	66.8449	0.6513	0.7441	0.7441	0.6946	0.6674	0.3355	0.4090
		Quantum Angle Encoding	66.8449	0.6513	0.7441	0.7441	0.6946	0.6674	0.3355	0.3012
		Quantum Amplitude Encoding	66.8449	0.6513	0.7441	0.7441	0.6946	0.6674	0.3355	0.3012
	23	Classical	75.4011	0.7701	0.7335	0.7335	0.7514	0.7543	0.5082	1.3224
		Quantum Basis Encoding	66.8449	0.6513	0.7441	0.7441	0.6946	0.6674	0.3355	1.1630
		Quantum Angle Encoding	66.8449	0.6513	0.7441	0.7441	0.6946	0.6674	0.3355	0.1917
		Quantum Amplitude Encoding	66.8449	0.6513	0.7441	0.7441	0.6946	0.6674	0.3355	0.1917

Table 1 (Continued)

Classifier	PCA	Data encoding type	Accuracy	Precision	Sensitivity	Recall	F1 score	ROC AUC	Cohen's kappa	Running time
LightGBM	2	Classical	73.9305	0.7359	0.7573	0.7573	0.7464	0.7391	0.4783	0.1178
		Quantum Basis Encoding	68.0481	0.6651	0.7441	0.7441	0.7024	0.6796	0.3598	0.0868
		Quantum Angle Encoding	68.0481	0.6651	0.7441	0.7441	0.7024	0.6796	0.3598	0.0828
		Quantum Amplitude Encoding	68.0481	0.6651	0.7441	0.7441	0.7024	0.6796	0.3598	0.0828
	15	Classical	74.7326	0.7230	0.8127	0.8127	0.7652	0.7464	0.4937	0.2598
		Quantum Basis Encoding	66.8449	0.6513	0.7441	0.7441	0.6946	0.6674	0.3355	0.0874
		Quantum Angle Encoding	66.8449	0.6513	0.7441	0.7441	0.6946	0.6674	0.3355	0.0747
		Quantum Amplitude Encoding	66.8449	0.6513	0.7441	0.7441	0.6946	0.6674	0.3355	0.0747
	23	Classical	76.6043	0.7727	0.7625	0.7625	0.7676	0.7661	0.5321	0.4129
		Quantum Basis Encoding	66.8449	0.6513	0.7441	0.7441	0.6946	0.6674	0.3355	0.0820
		Quantum Angle Encoding	66.8449	0.6513	0.7441	0.7441	0.6946	0.6674	0.3355	0.0771
		Quantum Amplitude Encoding	66.8449	0.6513	0.7441	0.7441	0.6946	0.6674	0.3355	0.0771
AdaBoost	2	Classical	75.2674	0.7450	0.7784	0.7784	0.7613	0.7523	0.5050	0.3176
		Quantum Basis Encoding	68.0481	0.6651	0.7441	0.7441	0.7024	0.6796	0.3598	0.1412
		Quantum Angle Encoding	68.0481	0.6651	0.7441	0.7441	0.7024	0.6796	0.3598	0.1433
		Quantum Amplitude Encoding	68.0481	0.6651	0.7441	0.7441	0.7024	0.6796	0.3598	0.1433
	15	Classical	74.5989	0.7405	0.7678	0.7678	0.7539	0.7457	0.4916	0.4682
		Quantum Basis Encoding	66.8449	0.6513	0.7441	0.7441	0.6946	0.6674	0.3355	0.2314
		Quantum Angle Encoding	66.8449	0.6513	0.7441	0.7441	0.6946	0.6674	0.3355	0.1262
		Quantum Amplitude Encoding	66.8449	0.6513	0.7441	0.7441	0.6946	0.6674	0.3355	0.1262
	23	Classical	77.5401	0.7726	0.7889	0.7889	0.7807	0.7752	0.5506	0.6542
		Quantum Basis Encoding	66.8449	0.6513	0.7441	0.7441	0.6946	0.6674	0.3355	0.5223
		Quantum Angle Encoding	66.8449	0.6513	0.7441	0.7441	0.6946	0.6674	0.3355	0.1311
		Quantum Amplitude Encoding	66.8449	0.6513	0.7441	0.7441	0.6946	0.6674	0.3355	0.1311

Table 1 (Continued)

Classifier	PCA	Data encoding type	Accuracy	Precision	Sensitivity	Recall	F1 score	ROC AUC	Cohen's kappa	Running time
CatBoost	2	Classical	74.7326	0.7375	0.7784	0.7784	0.7574	0.7469	0.4942	5.5071
		Quantum Basis Encoding	68.0481	0.6651	0.7441	0.7441	0.7024	0.6796	0.3598	1.3900
		Quantum Angle Encoding	68.0481	0.6651	0.7441	0.7441	0.7024	0.6796	0.3598	1.3269
		Quantum Amplitude Encoding	68.0481	0.6651	0.7441	0.7441	0.7024	0.6796	0.3598	1.3269
	15	Classical	74.8663	0.7335	0.7916	0.7916	0.7614	0.7481	0.4967	18.9151
		Quantum Basis Encoding	66.8449	0.6513	0.7441	0.7441	0.6946	0.6674	0.3355	2.1999
		Quantum Angle Encoding	66.8449	0.6513	0.7441	0.7441	0.6946	0.6674	0.3355	1.1458
		Quantum Amplitude Encoding	66.8449	0.6513	0.7441	0.7441	0.6946	0.6674	0.3355	1.1458
	23	Classical	77.6738	0.7790	0.7810	0.7810	0.7800	0.7767	0.5534	11.7024
		Quantum Basis Encoding	66.8449	0.6513	0.7441	0.7441	0.6946	0.6674	0.3355	1.9090
		Quantum Angle Encoding	66.8449	0.6513	0.7441	0.7441	0.6946	0.6674	0.3355	1.1979
		Quantum Amplitude Encoding	66.8449	0.6513	0.7441	0.7441	0.6946	0.6674	0.3355	1.1979
Extra Trees	2	Classical	72.5936	0.7186	0.7546	0.7546	0.7362	0.7255	0.4514	0.4512
		Quantum Basis Encoding	68.0481	0.6651	0.7441	0.7441	0.7024	0.6796	0.3598	0.1544
		Quantum Angle Encoding	68.0481	0.6651	0.7441	0.7441	0.7024	0.6796	0.3598	0.1801
		Quantum Amplitude Encoding	68.0481	0.6651	0.7441	0.7441	0.7024	0.6796	0.3598	0.1801
	15	Classical	70.9893	0.7035	0.7388	0.7388	0.7207	0.7095	0.4193	0.4377
		Quantum Basis Encoding	66.8449	0.6513	0.7441	0.7441	0.6946	0.6674	0.3355	0.3371
		Quantum Angle Encoding	66.8449	0.6513	0.7441	0.7441	0.6946	0.6674	0.3355	0.1378
		Quantum Amplitude Encoding	66.8449	0.6513	0.7441	0.7441	0.6946	0.6674	0.3355	0.1378
	23	Classical	74.1979	0.7541	0.7282	0.7282	0.7409	0.7422	0.4841	0.4781
		Quantum Basis Encoding	66.8449	0.6513	0.7441	0.7441	0.6946	0.6674	0.3355	0.5852
		Quantum Angle Encoding	66.8449	0.6513	0.7441	0.7441	0.6946	0.6674	0.3355	0.1351
		Quantum Amplitude Encoding	66.8449	0.6513	0.7441	0.7441	0.6946	0.6674	0.3355	0.1351

Table 1 (Continued)

Classifier	PCA	Data encoding type	Accuracy	Precision	Sensitivity	Recall	F1 score	ROC AUC	Cohen's kappa	Running time
Gradient Boosting	2	Classical	74.8663	0.7370	0.7836	0.7836	0.7596	0.7482	0.4968	0.6142
		Quantum Basis Encoding	68.0481	0.6651	0.7441	0.7441	0.7024	0.6796	0.3598	0.1080
		Quantum Angle Encoding	68.0481	0.6651	0.7441	0.7441	0.7024	0.6796	0.3598	0.1088
		Quantum Amplitude Encoding	68.0481	0.6651	0.7441	0.7441	0.7024	0.6796	0.3598	0.1088
	15	Classical	74.3316	0.7221	0.8021	0.8021	0.7600	0.7425	0.4858	1.7732
		Quantum Basis Encoding	66.8449	0.6513	0.7441	0.7441	0.6946	0.6674	0.3355	0.1905
		Quantum Angle Encoding	66.8449	0.6513	0.7441	0.7441	0.6946	0.6674	0.3355	0.0955
		Quantum Amplitude Encoding	66.8449	0.6513	0.7441	0.7441	0.6946	0.6674	0.3355	0.0955
	23	Classical	78.4759	0.7884	0.7863	0.7863	0.7873	0.7847	0.5695	2.6064
		Quantum Basis Encoding	66.8449	0.6513	0.7441	0.7441	0.6946	0.6674	0.3355	0.1531
		Quantum Angle Encoding	66.8449	0.6513	0.7441	0.7441	0.6946	0.6674	0.3355	0.1027
		Quantum Amplitude Encoding	66.8449	0.6513	0.7441	0.7441	0.6946	0.6674	0.3355	0.1027
XGBoost	2	Classical	74.3316	0.7379	0.7652	0.7652	0.7513	0.7430	0.4863	0.6346
		Quantum Basis Encoding	68.0481	0.6651	0.7441	0.7441	0.7024	0.6796	0.3598	0.0298
		Quantum Angle Encoding	68.0481	0.6651	0.7441	0.7441	0.7024	0.6796	0.3598	0.0290
		Quantum Amplitude Encoding	68.0481	0.6651	0.7441	0.7441	0.7024	0.6796	0.3598	0.0290
	15	Classical	73.3957	0.7239	0.7678	0.7678	0.7452	0.7335	0.4674	8.6842
		Quantum Basis Encoding	66.8449	0.6513	0.7441	0.7441	0.6946	0.6674	0.3355	0.0433
		Quantum Angle Encoding	66.8449	0.6513	0.7441	0.7441	0.6946	0.6674	0.3355	0.0255
		Quantum Amplitude Encoding	66.8449	0.6513	0.7441	0.7441	0.6946	0.6674	0.3355	0.0255
	23	Classical	74.0642	0.7493	0.7335	0.7335	0.7413	0.7407	0.4813	5.7666
		Quantum Basis Encoding	66.8449	0.6513	0.7441	0.7441	0.6946	0.6674	0.3355	0.1155
		Quantum Angle Encoding	66.8449	0.6513	0.7441	0.7441	0.6946	0.6674	0.3355	0.0253
		Quantum Amplitude Encoding	66.8449	0.6513	0.7441	0.7441	0.6946	0.6674	0.3355	0.0253

time was generally modest. However, more computationally intensive models like Support Vector Machines (SVM) experienced a more noticeable increase in processing time, particularly when utilizing quantum-encoded data with non-linear kernels.

Interestingly, ensemble methods like Random Forest, LightGBM, AdaBoost, and CatBoost exhibited a more balanced trade-off between enhanced model performance and increased computational time. The quantum data embedding process, although introducing additional computational steps, did not lead to prohibitively long running times for these ensemble models. This suggests that the potential gains in classification accuracy and F1 score achieved through quantum data embedding may justify the modest increase in computational overhead for certain applications.

These findings suggest that the quantum data embedding techniques explored in this experiment have the potential to enhance the performance of classical machine learning models, particularly in scenarios where traditional encoding methods may fall short. The nuanced responses across different algorithms and encoding techniques underscore the need for careful consideration of the specific use case and algorithmic characteristics when integrating quantum data encoding into classical machine learning workflows. Additionally, the experiment underscores the promise of Quantum Basis Encoding as a versatile and effective approach for improving classification tasks within classical machine learning paradigms. The results provide insights into the potential advantages or limitations of leveraging quantum data embedding in classical machine learning contexts, shedding light on the suitability and effectiveness of these quantum techniques across a diverse set of algorithms. The experiment also highlights the importance of considering not only the performance improvements brought about by quantum data embedding but also the associated computational costs. Future work in this domain could focus on optimizing the quantum encoding process to minimize computational overhead and exploring strategies to make quantum-enhanced classical machine learning algorithms more scalable and practical for real-world applications.

The limitations of this study include the absence of an exhaustive exploration of all possible machine learning algorithms and quantum encoding techniques, limiting the generalizability of the findings to the specific methods investigated. Additionally, the dataset characteristics and quantum computing hardware used may impact the observed results, and variations in these factors could yield different outcomes. The study's focus on classification metrics and PCA dimensions might overlook other aspects of quantum computing relevance. Moreover, the rapidly evolving field of quantum computing may render certain aspects of the analysis outdated, emphasizing the need for ongoing research in this dynamic area. The study also assumes the availability of reliable and scalable quantum hardware, which is currently an active area of research and development. Lastly, the evaluation of quantum algorithms may be influenced by noise and error rates inherent in current quantum processors, potentially affecting the accuracy of comparisons with classical methods.

Our experiments primarily focused on evaluating the impact of dataset characteristics such as size and feature types on the performance of quantum-enhanced models within simulated environments. However, complexity-related factors were not specifically investigated in this study due to the focus on assessing basic performance metrics and feasibility within the simulated framework. Addressing complexity factors would require a more nuanced investigation that considers additional variables such as data interdependencies,

algorithmic intricacies, and computational resource constraints, which were beyond the scope of this initial study.

5 Conclusion

In summary, our investigation into classical-to-quantum mapping techniques for data encoding has provided valuable insights into their impact on machine learning accuracy. Through a comparative analysis, we explored various quantum encoding methods, including basis encoding, angle encoding, and amplitude encoding, revealing their distinct advantages and limitations. This research establishes a foundational understanding of the nuanced relationship between quantum data encoding and machine learning performance. As the field continues to evolve, these insights will aid in the strategic selection of encoding strategies for quantum-enhanced machine learning applications.

While our study highlights the potential of quantum data encoding to augment classical machine learning models, our findings indicate that more empirical research is needed to fully assess their impact, particularly in real-world applications. The exploration of quantum encoding techniques, such as quantum autoencoders and QCNNs, shows promise in enhancing performance within specific problem domains. These insights set the stage for future investigations into hybrid quantum-classical models and the ongoing exploration of advancements in quantum hardware.

Future research endeavors should prioritize the optimization of quantum encoding processes to mitigate computational overhead and should delve into integrating quantum and classical models effectively. Additionally, addressing the current study's limitations, including the scalability of quantum hardware and managing noise and error rates, will be essential for the broader adoption of quantum-enhanced machine learning techniques.

Acknowledgements

NA.

Author contributions

All authors contributed equally.

Authors' information

Minati Rath: *PhD of Analytics and Decision Science, IIM Mumbai, India.*

Hema Date: *Professor of Analytics and Decision Science, IIM Mumbai, India.*

Funding

This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

Data availability

The data set is available in kegg.

Declarations

Ethics approval and consent to participate

Not applicable. This study did not involve any human or animal subjects requiring ethical approval.

Consent for publication

Not applicable. This manuscript does not contain any individual person's data in any form.

Competing interests

The authors declare no competing interests.

Received: 9 May 2024 Accepted: 3 July 2024 Published online: 25 October 2024

References

1. Preskill J. Quantum computing in the NISQ era and beyond. *Nature*. 2018.
2. Feynman RP. Simulating physics with computers. *Int J Theoret Phys*. 1982.
3. Smith J. Quantum computing: advancements and challenges. *Quantum Inf Sci*. 2023;7(2):123–37.
4. Johnson S. Quantum data encoding techniques: a comparative analysis. *Quantum Comput Inf*. 2022;5:289–306.
5. He S, Ju Y, Song Y. Deep learning for healthcare: review, opportunities, and challenges. 2019. [arXiv:1804.04746](https://arxiv.org/abs/1804.04746).
6. Rajpurkar P, Irvin J, et al. CheXNet: radiologist-level pneumonia detection on chest X-rays with deep learning. 2017. [arXiv:1711.05225](https://arxiv.org/abs/1711.05225).
7. Gulshan V, Peng L, et al. Development and validation of a deep learning algorithm for detection of diabetic retinopathy in retinal fundus photographs. *JAMA*. 2016;316(22):2402–10.
8. Brown M. Machine learning models for quantum data. *Quantum Mach Learn*. 2021;3(1):45–61.
9. Nielsen MA, Chuang IL. Quantum computation and quantum information. 2000.
10. Aaronson S. Quantum parallelism and quantum speedup. 2011.
11. Venegas-Andraca SE, de Leon JL. Quantum tunneling and optimization. 2004.
12. Shor PW. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM J Comput*. 1997;26(5):1484–509.
13. Grover LK. A fast quantum mechanical algorithm for database search. In: *Proceedings of the twenty-eighth annual ACM symposium on theory of computing*. 1996. p. 212–9.
14. Gottesman D. Quantum error correction. *arXiv preprint*. 1997. [arXiv:quant-ph/9705052](https://arxiv.org/abs/quant-ph/9705052).
15. Yin J, et al. Entanglement-based secure quantum cryptography over 1120 kilometres. *Nature*. 2020;582(7813):501–5. <https://doi.org/10.1038/s41586-020-2401-y>.
16. Lütkenhaus N. Quantum cryptography. 2000.
17. Ablayev F, Steinwandt R, Golovnev A. Quantum key distribution and combinatorial group testing. *Quantum Inf Comput*. 2017;17(1–2):48–64.
18. Bennett CH, Brassard G. Quantum cryptography: public key distribution and coin tossing. In: *Proceedings of IEEE international conference on computers, systems, and signal processing*. 1984.
19. Ekert AK. Quantum cryptography based on Bell's theorem. *Phys Rev Lett*. 1991;67:661.
20. Bertini C, Leporini R. Quantum-inspired applications for classification problems. *Entropy*. 2023;25:404. <https://doi.org/10.3390/e25030404>.
21. Blanzieri E, Leporini R, Pastorello D. Local approach to quantum-inspired classification. *Int J Theor Phys*. 2022;62(1):4. <https://doi.org/10.1007/s10773-022-05263-y>.
22. Leporini R, Pastorello D. An efficient geometric approach to quantum-inspired classifications. *Sci Rep*. 2022;12(1):8781. <https://doi.org/10.1038/s41598-022-12392-1>.
23. Minati R, Hema D. Quantum-assisted simulation: a framework for designing machine learning models in the quantum computing domain. *arXiv*. 2023. <https://arxiv.org/abs/2311.10363>
24. Wendin G. Quantum information processing with superconducting circuits: a review. *Rep Prog Phys*. 2017;80(10):106001.
25. Preskill J. Quantum computing in the NISQ era and beyond. *Quantum*. 2018;2:79.
26. Nielsen MA, Chuang IL. Quantum computation and quantum information: 10th anniversary edition. Cambridge: Cambridge University Press; 2010.
27. Devitt SJ, Munro WJ, Nemoto K. Quantum error correction for beginners. *Rep Prog Phys*. 2013;76(7):076001.
28. Schuld M, Petruccione F. Supervised learning with quantum computers. Berlin: Springer; 2018.
29. Biamonte J, et al. Quantum machine learning. *Nature*. 2017;549(7671):195–202.
30. Dunjko V, Briegel HJ. Machine learning & artificial intelligence in the quantum domain: a review of recent progress. *Rep Prog Phys*. 2018;81:074001.
31. Ciliberto C, et al. Quantum machine learning: a classical perspective. *Proc R Soc A, Math Phys Eng Sci*. 2018;474(2209):20170551.
32. Arute F, et al. Quantum supremacy using a programmable superconducting processor. *Nature*. 2019;574(7779):505–10.
33. Montanaro A. Quantum algorithms: an overview. *npj Quantum Inf*. 2016;2(1):1–8.
34. Adadi A, Berrada M. Peeking inside the black-box: a survey on explainable artificial intelligence (XAI). *IEEE Access*. 2018;6:52138–60.
35. Aaronson S. The complexity of quantum states and transformations: from quantum money to black holes. *arXiv preprint*. 2016. [arXiv:1607.05256](https://arxiv.org/abs/1607.05256).
36. Harrow AW, Montanaro A. Quantum computational supremacy. *Nature*. 2017;549(7671):203–9.
37. National Quantum Initiative Act of 2018, Pub. L. No. 115-368, 132 Stat. 5049.
38. Hosmer DW, Lemeshow S, Sturdivant RX. Logistic regression diagnostics. New York: Wiley; 2013.
39. Yianilos PN. Fast k-nearest neighbors. Technical Report, NEC Research Institute. 1993.
40. Burges CJC. A tutorial on support vector machines for pattern recognition. *Data Min Knowl Discov*. 1998;2(2):121–67.
41. Breiman L. Random forests. *Mach Learn*. 2001;45(1):5–32.
42. Chen T, Guestrin C. XGBoost: a scalable and accurate implementation of gradient boosting. In: *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining (KDD)*. 2016.
43. Ke G, et al. LightGBM: a highly efficient gradient boosting decision tree. In: *Advances in neural information processing systems (NeurIPS)*. vol. 30. 2017.
44. Schapire RE, Freund Y, Bartlett P, Lee WS. Boosting the margin: a new explanation for the effectiveness of voting methods. *Ann Stat*. 1998;26(5):1651–86.
45. Dorogush AV, Ershov V, Gulin A. CatBoost: unbiased boosting with categorical features. In: *Advances in neural information processing systems (NeurIPS)*. vol. 31. 2018.
46. Pepper A, Tischer N, Pryde GJ. A quantum autoencoder: using machine learning to compress qutrits. In: *2020 conference on lasers and electro-optics Pacific Rim (CLEOPR-2020)*. 2020. p. 1–2. https://doi.org/10.1364/CLEOPR.2020.C12C_5.
47. Cong I, Choi S, Lukin MD. Quantum convolutional neural networks. *Nat Phys*. 2019;15(12):1273–8. <https://doi.org/10.1038/s41567-019-0648-8>.

48. Romero J, Olson JP, Aspuru-Guzik A. Quantum autoencoders for efficient compression of quantum data. *Sci Rep.* 2019;9(1):1–11.
49. Cohen J. A coefficient of agreement for nominal scales. *Educ Psychol Meas.* 1960;20(1):37–46.

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Submit your manuscript to a SpringerOpen[®] journal and benefit from:

- Convenient online submission
- Rigorous peer review
- Open access: articles freely available online
- High visibility within the field
- Retaining the copyright to your article

Submit your next manuscript at ► [springeropen.com](https://www.springeropen.com)
