Bachelor of Science in Electrical and Electronic Engineering
EEE 400 (January 2024): Thesis

Performance Analysis and Noise Impact of a Novel Quantum KNN Algorithm for Machine Learning

Submitted by

Asif Akhtab Ronggon
1906078

Supervised by

Dr. Md. Saifur Rahman
Professor

Department of Electrical and Electronic Engineering

Bangladesh University of Engineering and Technology

Dhaka, Bangladesh

March 2025

CANDIDATES' DECLARATION

This is to certify that the work presented in this thesis, titled, "Performance Analysis and Noise Impact of a Novel Quantum KNN Algorithm for Machine Learning", is the outcome of the investigation and research carried out by me under the supervision of Dr. Md. Saifur Rahman.

It is also declared that neither this thesis nor any part thereof has been submitted anywhere else for the award of any degree, diploma or other qualifications.

Asif Akhtab Ronggon
1906078

CERTIFICATION

This thesis titled, **“Performance Analysis and Noise Impact of a Novel Quantum KNN Algorithm for Machine Learning”**, submitted by the group as mentioned below has been accepted as satisfactory in partial fulfillment of the requirements for EEE 400: Project/Thesis course, and as the requirements for the degree B.Sc. in Electrical and Electronic Engineering in March 2025.

Group Members:

Asif Akhtab Ronggon

Supervisor:

Dr. Md. Saifur Rahman
Professor
Department of Electrical and Electronic Engineering
Bangladesh University of Engineering and Technology

ACKNOWLEDGEMENT

I wish to express my profound gratitude to my supervisor, Dr Md. Saifur Rahman, for his invaluable guidance, thoughtful encouragement, and consistent support throughout the course of my research. His expertise and constructive feedback have been pivotal to the successful completion of this work.

I would also like to extend my sincere appreciation to Bangladesh University of Engineering and Technology and the Department of Electrical and Electronics Engineering for fostering an intellectually stimulating environment and for providing the resources necessary to pursue this research.

Furthermore, I am deeply thankful to the development teams behind PennyLane and Qiskit for their exceptional tools and comprehensive educational resources. The insights and understanding gained from these platforms have been instrumental in advancing my knowledge of quantum computing and machine learning, as well as in shaping the outcomes of this thesis.

Finally, I am deeply indebted to my parents for their unconditional love, patience, and faith in me. Their support has been my greatest source of strength throughout this journey.

Asif Akhtab Ronggon

Dhaka
March 2025

Contents

CANDIDATES' DECLARATION **i**

CERTIFICATION **ii**

ACKNOWLEDGEMENT **iii**

List of Figures **vii**

List of Tables **ix**

List of Algorithms **x**

ABSTRACT **xi**

1 Introduction **1**

1.1 Background and Motivation 1
1.2 Research Objectives 2
1.3 Research Contribution 2
1.4 Organization of the Thesis 3

2 Literature Review on Quantum Machine Learning Algorithms **4**

3 Quantum Computing **7**

3.1 History of Quantum Computing 7
3.2 Why Quantum Computing over Classical Computing 8
3.3 Postulates of Quantum Mechanics 9
3.3.1 Postulate 1: State Space 9
3.3.2 Postulate 2: Evolution of Quantum States 9
3.3.3 Postulate 3: Measurement 10
3.3.4 Postulate 4: Composite Systems and Entanglement 10
3.4 Qubits 11
3.5 Physical Realization of Qubits 11
3.6 Bell's Inequality 13
3.7 Quantum Superposition 14

3.8 Quantum Entanglement 15
3.9 Quantum Gates 19
3.10 Quantum Measurement 30
3.11 The No-Cloning Theorem 31
3.12 Quantum Algorithms 32
3.12.1 Deutsch-Jozsa Algorithm 32
3.12.2 Grover's Algorithm 34
3.12.3 Shor's Algorithm 35
3.12.4 Quantum Approximate Optimization Algorithm (QAOA) 37
3.13 Application of Quantum Computing 39
3.13.1 Quantum Cryptography 40
3.13.2 Quantum Machine Learning 40
3.13.3 Quantum Optimization 40
3.13.4 Drug Discovery and Quantum Chemistry 40
3.13.5 Materials Science and Simulation 41
3.13.6 Artificial Intelligence and Data Analysis 41
3.13.7 Quantum Networking and Communication 41

4 Development of a Quantum K-Nearest Neighbors Classifier: From Theory to Implementation **42**
4.1 Introduction 42
4.2 Problem Defination 43
4.3 Theoretical Foundation 44
4.3.1 Classical K-Nearest Neighbors 44
4.3.2 Quantum Machine Learning (QML): An Emerging Paradigm 46
4.3.3 Quantum Gates 48
4.3.4 Classical vs Quantum Feature Space 50
4.3.5 Quantum State Encoding 52
4.3.6 Quantum Distance Measures 55
4.3.7 Feature Selection Using Chi-Square Test 57
4.4 Proposed Algorithm Design:Quantum K-Nearest Neighbors (QKNN) 58
4.4.1 Input Representation: Encoding Classical Data into Quantum States . . 59
4.4.2 Feature Extraction using Quantum Gates 61
4.4.3 Distance Calculation and Neighbor Selection 63
4.4.4 Advantages and Limitations of the Proposed Quantum k-NN Algorithm 65
4.5 Simulation or Implementation Details 67
4.6 Noise Model and Simulation in Q-KNN 68
4.6.1 Quantum Noise Channels 68
4.6.2 Implementation of Noise in the Quantum Circuit 70

4.7 Introduction of Repetition Code for Error Mitigation 71

5 Results and Discussion **75**

6 Conclusion and Suggestions for Future Work **81**
6.1 Conclusion 81
6.2 Suggestions for Future Work 82

References **83**

A Codes **89**
A.1 Classical Bit vs Qubit Figure Generation Code 89
A.2 KNN Methodology 90
A.3 Quantum Entanglemet Figure Generation Code 92
A.4 Quantum Feature Space Figure Generation 93
A.5 Quantum KNN accuracy graph generation code 94

List of Figures

3.1 Classical Bit vs Qubit . . . 11

3.2 **Bell Test Experiment**: A quantum source emits entangled photon pairs, which are measured by Alice and Bob at spatially separated locations. The correlations between their measurement outcomes violate Bell's inequality, proving the nonlocal nature of quantum entanglement. . . . 14

3.3 Quantum spin states representation: spin-up, spin-down, and their superposition state. . . . 15

3.4 Quantum Entanglement . . . 16

3.5 Bell State Circuits with Their Corresponding Outputs . . . 17

3.6 Comparison between classical and quantum measurement outcomes [1, 2]. . . . 31

3.7 Quantum Circuit Diagram for the Deutsch–Jozsa Algorithm, illustrating the initialization, oracle (balanced or constant), and interference via Hadamard transforms. . . . 34

3.8 Quantum Circuit Diagram for Grover's Algorithm: Oracle and Diffuser Implementation for Marking $|101\rangle$ . . . 35

3.9 Quantum Circuit Diagram for Shor's Algorithm (Period-Finding Subroutine) . . 37

3.10 Quantum Approximate Optimization Algorithm (QAOA) workflow: The quantum processor applies cost and mixing Hamiltonians, while a classical optimizer updates parameters iteratively . . . 39

4.1 KNN Methodology . . . 45

4.2 Classical Feature Space vs Quantum Feature Space . . . 51

4.3 Q-KNN Methodology . . . 59

4.4 Q-KNN architecture . . . 62

4.5 Qiskit - IBM's Open-Source Quantum Computing Framework. It is used for simulating quantum circuits and executing them on IBM Quantum hardware. . 67

4.6 PennyLane - A Quantum Machine Learning and Simulation Platform. It is used for quantum computing simulations, variational quantum algorithms, and integrating quantum computations with machine learning. . . . 68

4.7 Quantum circuits representing different noise models with increasing Noise probability. . . . 71

5.1 Confusion Matrix for Banknote Authentication dataset 77
5.2 Confusion Matrix for Iris dataset 77
5.3 Confusion Matrix for Breast Cancer Wisconsin dataset 77
5.4 Confusion Matrices for Different Datasets 77
5.5 Comparitive Performance of proposed Q-KNN and KNN 78
5.6 Impact of Noise Probability on Model Performance 79

List of Tables

3.1 Different Physical Elements for Qubit Encoding [3–5] 12

4.1 Error Syndrome Table for 3-Qubit Repetition Code 73

5.1 Basic confusion matrix for classification. 75
5.2 Comparison of Proposed Q-KNN and KNN Accuracies Across Datasets 76
5.3 Performance Metrics of Proposed Quantum-KNN (Q-KNN) Architecture . . . 76

List of Algorithms

1 Deutsch-Jozsa Algorithm 33
2 Grover's Algorithm for Unstructured Search 35
3 Shor's Algorithm for Integer Factorization 36
4 Quantum Approximate Optimization Algorithm (QAOA) 38
5 Classical k-NN Algorithm 46
6 Quantum K-Nearest Neighbors (QKNN) Algorithm 65

ABSTRACT

This thesis presents the development and analysis of a Quantum K-Nearest Neighbors (Q-KNN) algorithm designed to enhance classification tasks of real-world problems by leveraging the principles of quantum computing. Traditional K-Nearest Neighbors (KNN) algorithms face computational inefficiencies, especially when handling high-dimensional datasets. To address this, Q-KNN utilizes quantum state encoding, quantum gate operations and swap tests to efficiently compute similarity measures in quantum-enhanced feature spaces.The study evaluates Q-KNN against Classical KNN (C-KNN) on three datasets: Breast Cancer, Iris, and Bank Note Authentication. The results demonstrate that Q-KNN consistently outperforms or matches the accuracy of C-KNN, achieving an accuracy of 98.25% for Breast Cancer (vs. 92.98%), 100% for Iris (same as C-KNN), and 99.27% for Bank Note Authentication (vs. 98.55%). Additional performance metrics, including F1-score, AUC, precision, and recall, further validate the superior classification ability of Q-KNN.The study also investigates the impact of quantum noise on Q-KNN's performance. The results indicate a strong inverse correlation between noise probability and accuracy, with performance deteriorating significantly beyond a noise threshold of 0.3. The findings suggest that quantum error correction techniques, such as repetition encoding, can mitigate noise-induced performance degradation, preserving model accuracy even under moderately noisy quantum conditions.The superior classification performance of Q-KNN highlights the potential of quantum machine learning in high-stakes applications, including biomedical diagnostics and financial fraud detection. Future research may be carried out to explore the scalability of Q-KNN, its robustness against more complex datasets, and its implementation on real quantum hardware to further validate its practical viability.

Chapter 1

Introduction

1.1 Background and Motivation

Machine learning has witnessed remarkable advancements in recent years, enabling intelligent decision-making in various domains such as biomedical diagnostics, financial fraud detection, and pattern recognition[1,2,3]. Among the widely used machine learning algorithms, the k-Nearest Neighbors (KNN) classifier is a simple yet effective method for classification tasks due to its non-parametric nature and ease of implementation. However, classical KNN models face challenges when dealing with large, complex datasets, as they suffer from high computational costs, especially in high-dimensional feature spaces[4,5].

With the advent of quantum computing, researchers have explored the potential advantages of quantum-enhanced machine learning models to overcome computational limitations associated with classical approaches. Quantum computing leverages the principles of superposition, entanglement, and quantum parallelism to perform computations more efficiently than classical systems. These quantum properties allow for enhanced pattern recognition and improved classification accuracy in various applications. Recent advancements in quantum machine learning (QML) have introduced Quantum-KNN (Q-KNN) as a promising alternative to its classical counterpart.

This study investigates the effectiveness of Q-KNN in comparison to classical KNN (C-KNN) across three benchmark datasets: Breast Cancer, Iris, and Bank Note Authentication. By leveraging quantum computational principles, Q-KNN aims to enhance classification performance and reduce computational complexity, making it a potential breakthrough in the field of machine learning.

1.2 Research Objectives

The primary objectives of this research are:

- To develop and implement a **Quantum-KNN (Q-KNN) model** and compare its performance with the classical KNN algorithm.
- To evaluate the classification performance of Q-KNN across three benchmark datasets—Breast Cancer, Iris, and Bank Note Authentication.
- To analyze key performance metrics such as accuracy, precision, recall, F1-score, and area under the curve (AUC) to assess the effectiveness of Q-KNN.
- To investigate the impact of quantum noise on model performance and explore possible error mitigation techniques.
- To propose future directions for enhancing the scalability and robustness of Q-KNN in real-world applications.

1.3 Research Contribution

The contribution of this study lies in its potential contributions to both quantum computing and machine learning. The application of Q-KNN introduces an innovative approach to classification tasks, demonstrating the feasibility of quantum-enhanced models in real-world datasets. Specifically, this research:

- Showcases **the superiority of Q-KNN over classical KNN** in terms of accuracy and other performance metrics.
- Provides **empirical evidence of quantum advantages** in complex classification problems, particularly in biomedical and financial datasets.
- Highlights the **sensitivity of quantum machine learning models to quantum noise**, emphasizing the need for robust quantum error correction techniques.
- Explores the potential of **scaling Q-KNN to larger and more complex datasets**, paving the way for future research in quantum machine learning.

By addressing these aspects, this research contributes to the growing field of quantum-enhanced machine learning, offering valuable insights for further advancements in the integration of quantum computing with traditional machine learning models.

1.4 Organization of the Thesis

This thesis is organized into five chapters, each addressing key aspects of Quantum-KNN and its application in classification tasks.

Chapter 2: Literature Review – This chapter presents an in-depth review of existing literature on classical machine learning models, particularly K-Nearest Neighbors (KNN), and the emergence of quantum machine learning (QML). It discusses previous research on Quantum-KNN (Q-KNN) and highlights key advancements in quantum-enhanced classification techniques.

Chapter 3: Quantum Computing – This chapter introduces fundamental quantum computing concepts, including superposition, entanglement, quantum measurement, and quantum gates. Additionally, it explores the theoretical underpinnings of quantum algorithms and their applications in machine learning.

Chapter 4: Developing a Quantum K-Nearest Neighbors Classifier: From Theory to Implementation – This chapter outlines the design and implementation of the proposed Q-KNN model. It describes the dataset selection, preprocessing techniques, quantum circuit construction, and the classical-quantum hybrid framework. Furthermore, it details the evaluation metrics and experimental setup used to compare Q-KNN with classical KNN.

Chapter 5: Results and Discussion – This chapter presents the experimental results obtained from the application of Q-KNN and classical KNN across multiple datasets. It includes a comparative analysis of classification performance, visualization of confusion matrices, and an assessment of key metrics such as accuracy, precision, recall, F1-score, and AUC. The impact of quantum noise on model performance is also discussed, along with potential strategies for error mitigation.

Chapter 6: Conclusion and Future Work – The final chapter summarizes the key findings of this research and highlights the advantages of Q-KNN over classical KNN. It discusses the limitations of the study and provides directions for future research, including scalability improvements, robustness in noisy quantum environments, and the feasibility of implementing Q-KNN on real quantum hardware.

This structured organization ensures a logical flow of concepts, beginning with foundational knowledge and culminating in experimental validation and conclusions.

.

Chapter 2

Literature Review on Quantum Machine Learning Algorithms

The emerging field of quantum machine learning harnesses the principles of quantum computing and classical machine learning to enhance computational efficiency and enable novel applications. The unique potential of QML stems from its capacity to leverage quantum phenomena, such as superposition and entanglement, to solve specific computations and problems in polynomial or exponential time, a feat that is unattainable for classical computers under certain constraints [6].

Adapting classical machine learning approaches to the quantum realm is a fundamental aspect of quantum machine learning. QML algorithms have exhibited notable advantages over traditional classical methods in various tasks, such as classification, regression, clustering, and optimization, where computational constraints often impede classical approaches. In their seminal work, Schuld and Petruccione provide a comprehensive overview of diverse QML algorithms, encompassing Variational QML, Quantum Adiabatic Machine Learning, Quantum Neural Networks, Quantum Kernels, Probabilistic models, and Quantum Walk Models [7].The success of quantum machine learning will largely depend on the development of quantum algorithms to perform fundamental machine learning tasks on quantum computers. A range of quantum algorithms have been proposed to solve fundamental problems like linear algebra, optimization, and data analysis, which form the foundation of many machine learning algorithms. For instance,The Quantum Fourier Transform(QFT) can be used for signal processing, frequency domain analysis, and encoding features in quantum states. As Lloyd et al. describe, an algorithm based on QFT can reduce the complexity of different types of signal analysis and feature extraction in machine learning tasks [8].The HHL algorithm, developed by Harrow, Hassidim, and Lloyd, is designed to solve linear systems of equations exponentially faster than classical approaches. As a result, this quantum algorithm can be utilized to address machine learning tasks involving linear systems in a more efficient manner than existing classical techniques [9].Variational Quantum Algorithms have emerged as a practical solution for near-term quantum devices. These algorithms combine classical optimization techniques with quantum circuits to solve various mathematical and optimization problems. In the context of machine learning and data science, VQAs play a key role in building models such as classifiers, autoencoders, generative models, and quantum neural networks [10].Farhi and Neven (2018) developed a quantum neural network design that use quantum gates to execute operations similar to those of a conventional neural network. They showed that, under certain situations, QNNs might outperform traditional neural networks in terms of computing efficiency, notably in tasks like classification and regression [11]. The use of Variational Quantum Eigensolvers (VQE) and Quantum Approximate Optimization techniques (QAOA) to train quantum neural networks has yielded encouraging results, as both techniques use quantum circuits to approximate solutions to optimization problems [12].Classical kernel methods map data into high-dimensional feature spaces, allowing linear classifiers to solve complex nonlinear problems. However, as datasets become more intricate and high-dimensional, computing these kernels can be computationally intensive. The quantum kernel presents a novel approach by harnessing the exponentially large Hilbert space of quantum systems to perform feature mapping and inner product computations more efficiently. Havlíček's work introduced quantum-enhanced feature spaces, demonstrating how quantum circuits could efficiently compute inner products for kernel estimation. Their findings indicated that quantum kernels could outperform classical kernels in specific cases by leveraging quantum entanglement and superposition [13]. Building upon this, Schuld and Killoran further investigated quantum embeddings for machine learning, highlighting the theoretical advantages of quantum kernels for high-dimensional feature spaces that are inaccessible to classical systems [14].Benedetti and colleagues developed a variational quantum algorithm for Quantum Support Vector Machines, where the kernel matrix was calculated using quantum circuits while the SVM training was performed classically. The authors demonstrated that their variational approach could optimize quantum circuits for kernel computation and evaluated their method on quantum simulators and hardware. Although the technique exhibited promise for both small and large datasets, the researchers acknowledged the persisting practical challenges, such as noise and error correction, that require further investigation [15].A quantum K-Nearest Neighbor classification algorithm leveraging the Mahalanobis distance is proposed, integrating the classical KNN approach with quantum computing principles to address supervised classification challenges in machine learning [16].The proposed Polar distance metric demonstrates enhanced scalability and robustness compared to the Euclidean distance, offering potential for the widespread utilization of Quantum K-Nearest Neighbor classification in real-world applications [17].The authors propose a novel quantum circuit that incorporates quantum phase estimation, controlled rotation, and inverse phase estimation techniques to quantize the process. This enables their algorithm to determine the optimal K value and identify the nearest neighbors of the testing data [18].Yue Ruan and colleagues develop a quantum algorithm for the K-nearest neighbors classification task. The proposed approach leverages the Hamming distance metric, which enables the algorithm's performance to scale solely with the dimensionality of the feature space while maintaining high predictive accuracy [19].The research presented a hybrid system for fake news detection, incorporating a quantum k-nearest neighbors machine learning model and Genetic and Evolutionary Feature Selection, which attained a top accuracy of 87.12% [20].The paper demonstrates that the investigation of a diverse set of thirteen datasets reveals the quantum k-nearest neighbors algorithms utilize the proposed subroutine to achieve at least a 50% reduction in the required number of qubits, while maintaining a comparable level of overall performance [21].

Chapter 3

Quantum Computing

Quantum computing is a field of computing that utilizes the principles of quantum mechanics, where quantum bits, or *qubits*, can exist in multiple states simultaneously due to superposition and be entangled with one another. This allows quantum computers to process information in parallel, potentially solving certain complex problems much faster than classical computers.

3.1 History of Quantum Computing

The origins of quantum computing can be traced back to the early 20th century when pioneering physicists, such as Max Planck, Niels Bohr, and Erwin Schrödinger, laid the groundwork for quantum mechanics [3]. The field emerged at the intersection of quantum mechanics and computational theory. In the 1980s, researchers like Richard Feynman and Yuri Manin observed the limitations of classical computers in efficiently simulating quantum systems, suggesting the need for a fundamentally different computational approach [22, 23]. This idea was formalized by David Deutsch in 1985, who proposed the concept of a universal quantum computer based on quantum principles like superposition and entanglement [24]. The field gained significant momentum in 1994 when Peter Shor developed an algorithm for efficiently factoring large integers, demonstrating the potential of quantum computers to outperform classical counterparts in critical tasks such as cryptography [25]. Early experimental work, including the implementation of quantum algorithms using nuclear magnetic resonance systems in the late 1990s, showcased the initial feasibility of the technology [26].

By 2001, IBM and Stanford researchers successfully demonstrated Shor's algorithm on a 7-qubit quantum computer, marking a significant milestone in hardware development [27]. The 2000s witnessed substantial progress in quantum hardware, particularly with the advent of superconducting qubits and trapped ions, technologies that enabled more stable and scalable quantum systems [5]. Companies like D-Wave Systems began exploring quantum annealing, while academia and industry worked on gate-based quantum computing [28]. A landmark achievement came in 2019, when Google claimed ”quantum supremacy” by performing a task on their Sycamore processor in 200 seconds—a feat that would take the most advanced classical supercomputers thousands of years [4].

Ongoing quantum computing research focuses on overcoming critical challenges, including quantum error correction, qubit coherence, and the integration of quantum processors with classical systems in hybrid computational models [29]. These advancements are crucial for transitioning from noisy intermediate-scale quantum (NISQ) devices to fault-tolerant quantum computers capable of solving real-world problems in cryptography, optimization, finance, computation, data analysis, material science, and artificial intelligence [30].

3.2 Why Quantum Computing over Classical Computing

Quantum computing has demonstrated remarkable promise and potential in solving specialized complex problems, outperforming classical computing in various capacities. The unique quantum mechanical properties of superposition and entanglement empower quantum computers to perform certain computations exponentially faster than the most advanced classical algorithms. For instance, quantum computers can tackle optimization problems and address NP-hard challenges more efficiently than classical systems. Algorithms like the Quantum Approximate Optimization Algorithm can rapidly approximate solutions for complex optimization problems in areas such as logistics, finance, and scheduling [31]. Quantum algorithms like Grover’s and Shor’s can provide quadratic speedups for unsorted database search and polynomial-time factorization of large numbers, respectively [32]. The superposition and entanglement characteristics of quantum systems make them exceptionally suited for problems where parallel processing is advantageous, such as quantum cryptography, quantum simulation, and quantum machine learning [33] [34]. Quantum computers excel at simulating quantum phenomena, leading to breakthroughs in diverse fields like drug discovery, material science, and biological system understanding [35]. Quantum versions of linear algebra and sampling methods can offer significant speed-ups in certain machine learning applications [36]. Additionally, quantum computers have the potential to perform certain calculations with reduced energy consumption compared to classical supercomputers, and quantum-secured cloud computing can provide heightened security against quantum-enabled cyber threats [37] [38]. Quantum sensors can also achieve higher precision than classical sensors, with applications in areas like imaging and medical diagnostics [39]. In summary, quantum computing promises to revolutionize a wide range of fields, from chemistry and physics to cryptography and materials science, by enabling new computational approaches that surpass the limitations of classical computers.

3.3 Postulates of Quantum Mechanics

Quantum mechanics is governed by several fundamental postulates that establish the framework for understanding quantum systems. These postulates form the theoretical foundation of quantum computing [1].

3.3.1 Postulate 1: State Space

A quantum system is fully described by a state vector, which resides in a complex Hilbert space. The state vector provides complete information about the system.

Mathematically, a quantum state is expressed as a linear combination of basis states:

$$|\psi\rangle = \sum_i c_i |\phi_i\rangle, \tag{3.1}$$

where $|\phi_i\rangle$ are basis vectors and c_i are complex coefficients satisfying the normalization condition [40]:

$$\langle\psi|\psi\rangle = 1. \tag{3.2}$$

In quantum computing, qubits (quantum bits) are represented as [1]:

$$|0\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad |1\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix}. \tag{3.3}$$

A general qubit state is given by:

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle, \tag{3.4}$$

where α and β are complex numbers satisfying $|\alpha|^2 + |\beta|^2 = 1$.

3.3.2 Postulate 2: Evolution of Quantum States

The time evolution of a quantum system is determined by the Schrödinger equation:

$$i\hbar \frac{d}{dt}|\psi(t)\rangle = \hat{H}|\psi(t)\rangle, \tag{3.5}$$

where $\hat{H}$ represents the Hamiltonian operator, which governs the total energy of the system [41].

For quantum gates in quantum computing, the evolution is characterized by unitary operators

U such that:

$$|\psi'\rangle = U|\psi\rangle, \tag{3.6}$$

where U satisfies $U^\dagger U = I$, ensuring reversibility [1].

3.3.3 Postulate 3: Measurement

Quantum measurement is described by a set of measurement operators M_i that satisfy the completeness relation [42]:

$$\sum_i M_i^\dagger M_i = I. \tag{3.7}$$

The probability of obtaining outcome i when measuring a state $|\psi\rangle$ is given by:

$$P(i) = \langle\psi|M_i^\dagger M_i|\psi\rangle. \tag{3.8}$$

Upon measurement, the system collapses to the state:

$$|\psi'\rangle = \frac{M_i|\psi\rangle}{\sqrt{P(i)}}. \tag{3.9}$$

In quantum computing, measurements are commonly performed in the computational basis $\{|0\rangle, |1\rangle\}$.

3.3.4 Postulate 4: Composite Systems and Entanglement

For composite quantum systems, the state space is given by the tensor product of the individual subsystems:

$$\mathcal{H} = \mathcal{H}_A \otimes \mathcal{H}_B. \tag{3.10}$$

This formulation gives rise to quantum entanglement, which plays a crucial role in quantum algorithms such as superdense coding and quantum teleportation. A maximally entangled Bell state is expressed as [1]:

$$|\Phi^+\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle). \tag{3.11}$$

The postulates of quantum mechanics establish the fundamental principles governing quantum states, evolution, and measurement. These principles translate directly into quantum computing, where unitary transformations, superposition, and entanglement enable powerful algorithms. Understanding the mathematical framework linking quantum mechanics to quantum computing is essential for leveraging quantum computational advantages.

3.4 Qubits

Information is fundamental to the digital age, and as technology advances, the demand for efficient and powerful information processing grows even more critical [43]. Classical computing processes information using bits that are either 0 or 1. Quantum computing introduces a novel approach to information processing that can surpass classical computing. Quantum computing employs qubits as the fundamental units for storing and processing information. Unlike classical bits, qubits can represent 0 and 1 simultaneously, enabling parallel quantum computing. Figure 3.1 represents the difference between the classical bit and the quantum qubit.

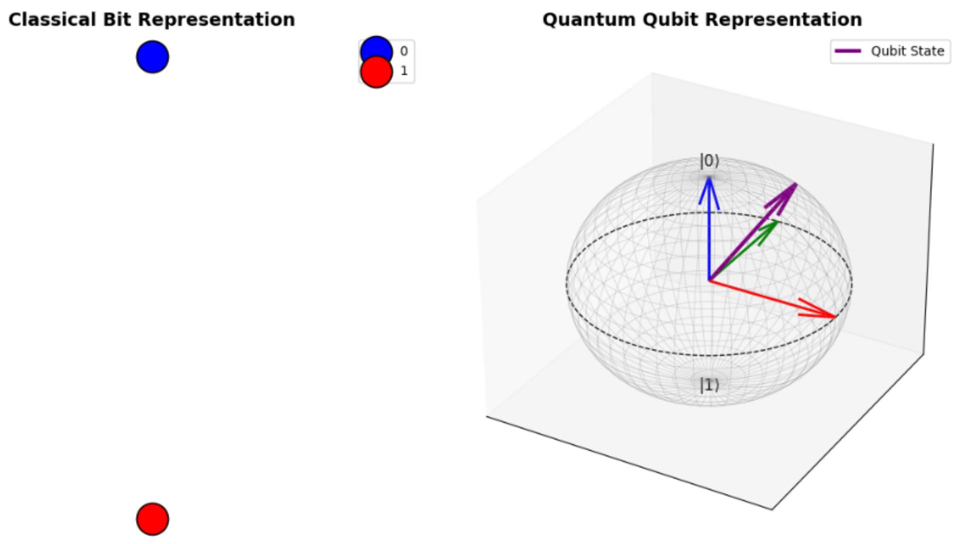


Figure 3.1: Classical Bit vs Qubit

3.5 Physical Realization of Qubits

In his work, David P. DiVincenzo outlines the essential criteria for the physical embodiment of the fundamental qubit unit in quantum computing. He delineated five critical requirements for the physical implementation of qubits in quantum computing.The requirements include a scalable physical system that can accommodate additional qubits without compromising performance, proper initialization of qubits before the computational process, a coherence time exceeding the gate operation time to prevent the qubit from losing its state before the gate operation completes, a universal set of quantum gates to enable the execution of any quantum operation, and a qubit-specific measurement capability to obtain the accurate result state after computation [44].

Researchers have explored various physical systems to realize qubits that meet the criteria outlined by DiVincenzo, with varying degrees of success. These include the use of individual atoms, electrons, photons, and superconducting electrical circuits as the foundation for qubits. Each system possesses distinct characteristics and offers unique advantages over the others.

Table 3.1: Different Physical Elements for Qubit Encoding [3–5]

Physical Element	Encoding	Principle
Photon [45]	**Polarization**	The state of a qubit can be encoded using the polarization of photons. Specifically, horizontal polarization can represent $\|0\rangle$, while vertical polarization can represent $\|1\rangle$.
	Path Encoding	The spatial paths of a photon can also be used to encode qubit states, where two different paths represent the $\|0\rangle$ and $\|1\rangle$ states.
	Time Bin Encoding	Qubits can also be encoded by the arrival time of a photon, where photons arriving in distinct time bins correspond to the $\|0\rangle$ and $\|1\rangle$ qubit states.
Electron [46]	**Spin**	Qubits are encoded using the spin states of electrons, with spin-up representing the $\|0\rangle$ state and spin-down representing the $\|1\rangle$ state.
Quantum Dot [47]	**Dot Spin**	The spin state of an electron confined within a quantum dot can be leveraged to encode the $\|0\rangle$ and $\|1\rangle$ qubit states.
Josephson Junction [48]	**Superconducting Charge**	A charge qubit employs a Cooper-pair box and a Josephson junction to encode the quantum $\|0\rangle$ and $\|1\rangle$ states, leveraging Cooper pairs to enable tunneling and superposition phenomena.
	Superconducting Flux	A flux qubit encodes the $\|0\rangle$ and $\|1\rangle$ qubit states by leveraging magnetic flux quantization within a superconducting loop. The two distinct energy minima correspond to different directions of the supercurrent, enabling superposition states and quantum operations.
	Superconducting Phase	The phase qubit encodes the quantum $\|0\rangle$ and $\|1\rangle$ states by utilizing the phase difference across a Josephson junction, with the energy levels determined by the ratio of Josephson energy to charging energy.
Trapped Ion [49]	**Atomic Energy Level**	Qubits can be encoded using the energy levels of individual ions, where one energy level represents the $\|0\rangle$ state and another energy level represents the $\|1\rangle$ state.

3.6 Bell's Inequality

Bell's inequality, proposed by John Bell in 1964 [50], is a fundamental theorem in quantum mechanics that challenges the principles of local realism, which assume that physical properties of particles are determined by pre-existing hidden variables and that information cannot travel faster than light. The inequality provides a testable condition to distinguish between classical physics and quantum mechanics by analyzing the statistical correlations between measurements of entangled particles. If quantum mechanics adheres to local realism, the correlations between measurements should satisfy Bell's inequality; however, quantum entanglement predicts stronger correlations that violate this classical bound [51].

The inequality is commonly formulated using the CHSH (Clauser-Horne-Shimony-Holt) version, which involves two observers, Alice and Bob, measuring entangled particles at spatially separated locations. Each observer independently chooses between two possible measurement settings, A_1, A_2 for Alice and B_1, B_2 for Bob, with each measurement producing outcomes of either $+1$ or -1. The correlation function between their results is expressed as:

$$S = E(A_1, B_1) + E(A_1, B_2) + E(A_2, B_1) - E(A_2, B_2) \tag{3.12}$$

where $E(A, B)$ represents the expectation value of the product of Alice's and Bob's measurement outcomes. In local hidden variable theories, $|S| \leq 2$, but quantum mechanics predicts that for maximally entangled states, $|S|$ can reach $2\sqrt{2} \approx 2.828$, thus violating Bell's inequality and confirming quantum nonlocality [52].Figure 3.2 represents Bell Test experiment identifying Nonlocal Correlations.

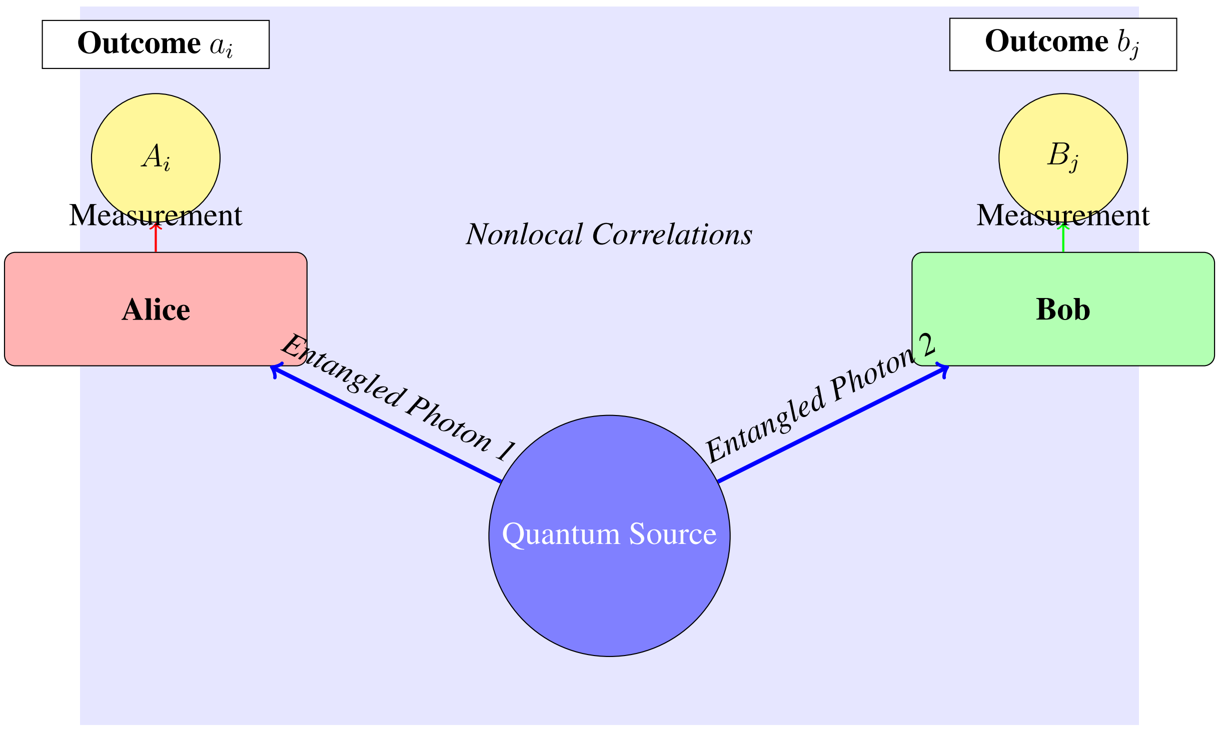


Figure 3.2: **Bell Test Experiment**: A quantum source emits entangled photon pairs, which are measured by Alice and Bob at spatially separated locations. The correlations between their measurement outcomes violate Bell's inequality, proving the nonlocal nature of quantum entanglement.

Experimental violations of Bell's inequality provide strong evidence that nature does not conform to classical local realism. Pioneering experiments, such as those conducted by Aspect et al. in 1982 [52], observed significant violations using photon polarization measurements. More recent loophole-free experiments, performed in 2015 using superconducting qubits and high-efficiency detectors, have reinforced these findings [53]. The implications of Bell's theorem extend beyond fundamental physics, with practical applications in quantum cryptography, particularly in quantum key distribution (QKD) protocols such as E91 [54], where the violation of Bell's inequality ensures secure communication against potential eavesdroppers. These results solidify the role of quantum entanglement in both foundational quantum mechanics and emerging quantum technologies.

3.7 Quantum Superposition

Quantum superposition, a foundational concept in quantum mechanics, refers to the capacity of a quantum system to occupy multiple states or configurations simultaneously, before measurement or observation. This implies that a quantum system can exist in a state that is a combination or "superposition" of various possible states, rather than being limited to a single definite state. Schrödinger's wave function postulate, a fundamental concept in quantum mechanics, gives rise to the notion of quantum superposition. This postulate describes a quantum system using a wave function $\psi(x,t)$, which encompasses all the possible states of the system. Quantum superposition asserts that if $\psi_1(x,t)$, $\psi_2(x,t)$, ..., $\psi_n(x,t)$ are valid wave functions corresponding to different states, the combined state can be expressed as a linear combination of these wave functions [55].

$$\psi(x,t) = \sum_{k=1}^{n} c_k \psi_k(x,t) \tag{3.13}$$

where c_k are complex coefficients.This understanding is crucial for grasping the counterintuitive behavior of quantum systems.Numerous experiments have been conducted to demonstrate this quantum phenomenon. For instance, the double-slit experiment has revealed that particles display wave-like interference patterns, implying they exist in a superposition of multiple paths [56] [57]. Furthermore, a beryllium ion has been trapped in a superposed state [58], and a piezoelectric tuning fork has been constructed such that it occupies a superposition of vibrating and non-vibrating states.Figure 3.3 represents quantum superposition of electron spin.

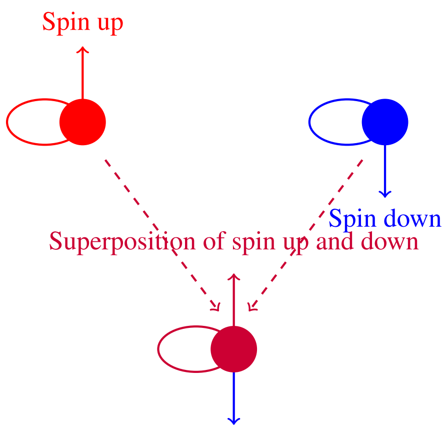


Figure 3.3: Quantum spin states representation: spin-up, spin-down, and their superposition state.

3.8 Quantum Entanglement

Quantum entanglement is a core and seminal principle in quantum mechanics. It describes a situation where two or more quantum systems become intrinsically interconnected, such that an action performed on the state of one system instantaneously affects the state of the other, regardless of their spatial separation. This implies that the quantum states of the interlinked systems cannot be independently characterized, and knowledge of the state of one system necessitates concurrent information about the other. This non-local attribute of entanglement gives rise to intriguing phenomena and holds significant implications across diverse fields, including quantum computing, quantum cryptography, and quantum teleportation [59].Figure 3.4 represents the quantum entanglement.

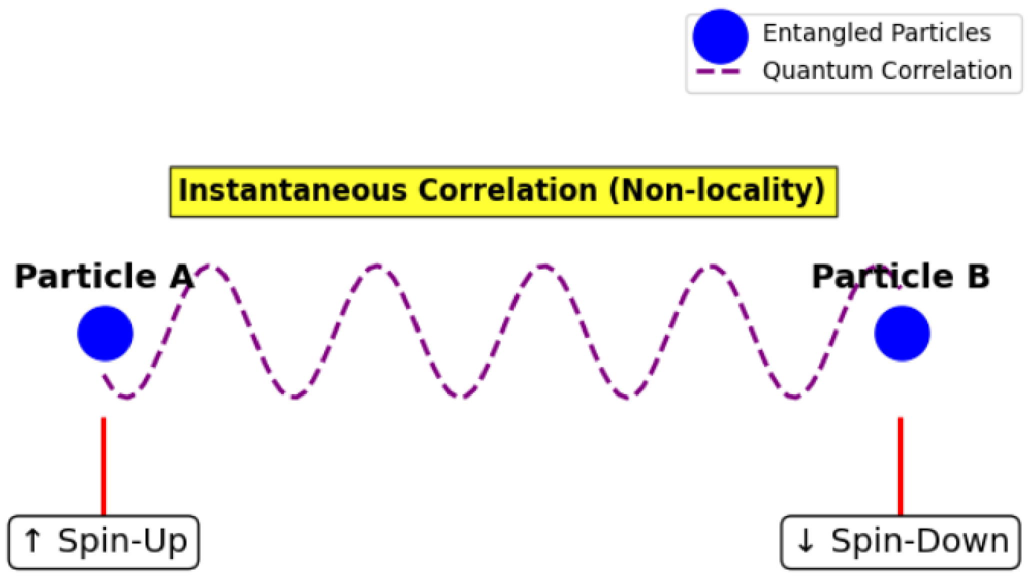


Figure 3.4: Quantum Entanglement

In the 1980s, Alain Aspect and his team conducted experiments that provided experimental confirmation of quantum entanglement, building upon Bell's Theorem. Their experiments demonstrated correlations between entangled particles that surpassed classical limits, addressing locality-related loopholes. Similarly, John Clauser's early work in the 1970s yielded the first experimental violations of Bell inequalities, while Anton Zeilinger's later research achieved entanglement over long distances, advancing applications in quantum teleportation and cryptography. In recognition of these foundational contributions, Clauser, Aspect, and Zeilinger were awarded the 2022 Nobel Prize in Physics. Their experiments firmly established quantum entanglement as a non-classical phenomenon, paving the way for technologies like device-independent quantum cryptography and global-scale quantum communication through satellite-linked entanglement.

In quantum mechanics, the quantum state of a particle is represented by a vector in a complex Hilbert space, $\mathcal{H}$. When two particles are in separate states $|\psi\rangle$ and $|\phi\rangle$, their combined state in the joint Hilbert space $\mathcal{H}_1 \otimes \mathcal{H}_2$ is given by the tensor product:

$$|\Psi_{\text{combined}}\rangle = |\psi\rangle \otimes |\phi\rangle \tag{3.14}$$

However, for an entangled system, the joint state cannot be expressed as a simple product of the individual states. Instead, the entangled state $|\Psi\rangle$ is a superposition of product states, described as:

$$|\Psi\rangle = \sum_{i,j} c_{ij} |i\rangle_1 \otimes |j\rangle_2 \tag{3.15}$$

where $|i\rangle_1$ and $|j\rangle_2$ are basis states for particles 1 and 2, and c_{ij} are complex coefficients [33]. The non-separable nature of the entangled state $|\Psi\rangle$ is the defining characteristic of quantum entanglement.Here are some common types of entangled states:

Bell States

Bell states are the most basic forms of quantum entanglement, consisting of two qubits. They provide an orthonormal basis for the two-qubit Hilbert space:

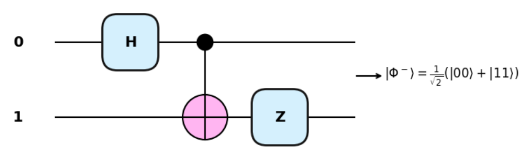
$$|\Phi^+\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle) \tag{3.16}$$

$$|\Phi^-\rangle = \frac{1}{\sqrt{2}}(|00\rangle - |11\rangle) \tag{3.17}$$

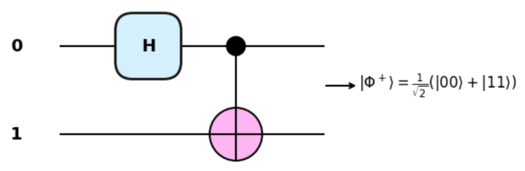
$$|\Psi^+\rangle = \frac{1}{\sqrt{2}}(|01\rangle + |10\rangle) \tag{3.18}$$

$$|\Psi^-\rangle = \frac{1}{\sqrt{2}}(|01\rangle - |10\rangle) \tag{3.19}$$

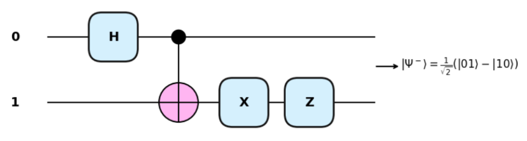
These are the maximally entangled state.Figure 3.5 represents the different bell state circuits.


0
1
H
Z
$\rightarrow |\Phi^-\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$

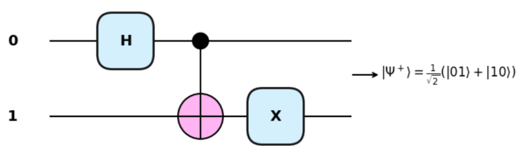
(a) $|\Phi^-\rangle = \frac{1}{\sqrt{2}}(|00\rangle - |11\rangle)$


0
1
H
$\rightarrow |\Phi^+\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$

(b) $|\Phi^+\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$


0
1
H
X
Z
$\rightarrow |\Psi^-\rangle = \frac{1}{\sqrt{2}}(|01\rangle - |10\rangle)$

(c) $|\Psi^-\rangle = \frac{1}{\sqrt{2}}(|01\rangle - |10\rangle)$


0
1
H
X
$\rightarrow |\Psi^+\rangle = \frac{1}{\sqrt{2}}(|01\rangle + |10\rangle)$

(d) $|\Psi^+\rangle = \frac{1}{\sqrt{2}}(|01\rangle + |10\rangle)$

Figure 3.5: Bell State Circuits with Their Corresponding Outputs

GHZ States

GHZ states are maximally entangled states that involve three or more qubits. The GHZ state of three qubits is as follows:

$$|GHZ\rangle = \frac{1}{\sqrt{2}}\left(|000\rangle + |111\rangle\right) \tag{3.20}$$

For n-qubits, the general GHZ state is:

$$|GHZ_n\rangle = \frac{1}{\sqrt{2}}\left(|0\rangle^{\otimes n} + |1\rangle^{\otimes n}\right) \tag{3.21}$$

W States

The W state is a three-qubit entangle state with less entanglement than the GHZ state, but it has the interesting property of remaining entangled even after a qubit is lost. This characteristic makes the state valuable in some applications.The W state for three qubits is as follows:

$$|W\rangle = \frac{1}{\sqrt{3}}\left(|001\rangle + |010\rangle + |100\rangle\right) \tag{3.22}$$

Bipartite Continuous-Variable Entangled States

For continuous-variable systems, an example of an entangled state is the two-mode squeezed vacuum state:

$$|\Psi\rangle = \sqrt{1-\lambda^2}\sum_{n=0}^{\infty} \lambda^n |n\rangle_1 |n\rangle_2 \tag{3.23}$$

where λ determines the degree of squeezing and $|n\rangle$ are Fock states.

Mixed Entangled States

Mixed states, such as Werner states, can also exhibit entanglement:

$$\rho_W = p|\Phi^-\rangle\langle\Phi^-| + (1-p)\frac{\mathbb{I}}{4} \tag{3.24}$$

where $|\Phi^-\rangle$ is a Bell state, p is a mixing parameter, and $\mathbb{I}/4$ is the maximally mixed state.

3.9 Quantum Gates

Conventional computing relies on digital logic gates that operate using classical binary digits of 0 and 1. These fundamental gates, including AND, OR, and NOT, manipulate these classical bits. In contrast, quantum computing leverages quantum gates, which are constructed upon the principles of quantum mechanics, to manipulate quantum bits or qubits instead of classical bits.

Identity Gate

The **Identity gate** is a single-qubit gate that acts as the quantum equivalent of a “no-operation” (NOP) in classical computing. Applying the Identity gate to a qubit leaves it in the same state. Although it may appear trivial, the Identity gate plays an important role in theoretical quantum mechanics and is often used in quantum circuits to represent time delays, to maintain dimensional consistency, or in constructing composite quantum operations.The Identity gate, denoted by I, is represented by the 2×2 identity matrix:

$$I = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \tag{3.25}$$

This matrix has eigenvalues of $+1$, which implies that it leaves any state vector unchanged. The Identity gate is a special case of rotation gates, where it corresponds to a 0-degree rotation around any axis.The Identity gate has the following effect on the computational basis states $|0\rangle$ and $|1\rangle$:

$$I\,|0\rangle = |0\rangle\,, \quad I\,|1\rangle = |1\rangle \tag{3.26}$$

Since the Identity gate does not alter the state of any qubit, applying it to a superposition state, such as $\alpha\,|0\rangle + \beta\,|1\rangle$, results in no change:

$$I(\alpha\,|0\rangle + \beta\,|1\rangle) = \alpha\,|0\rangle + \beta\,|1\rangle \tag{3.27}$$

This property makes the Identity gate a valuable tool in situations where a qubit’s state should remain stable over time or where no transformation is desired.On the Bloch sphere, the Identity gate corresponds to a 0-degree rotation around any axis, which leaves the position of the qubit’s state vector unchanged. In this sense, the Identity gate can be seen as a “do-nothing” transformation that maintains the original quantum state.While the Identity gate does not modify the qubit state, it is commonly used in quantum circuits for purposes such as:

1. Timing Control: Adding Identity gates can adjust the timing of specific operations.

2. Dimensional Consistency: Ensures that all qubits in a multi-qubit system have an operation applied to them, preserving matrix dimensionality.

3. Error Correction: Often used in fault-tolerant circuits and in combination with other gates to test for potential errors.

The Identity gate is also used in defining other operations. For example, in controlled gates, the Identity gate is applied to qubits that should remain unchanged.In a quantum circuit, the Identity gate is represented by the symbol I or is sometimes omitted entirely. For a qubit initialized in state $|\psi\rangle$, the application of the Identity gate is shown as:

$$|\psi\rangle \; -\!\boxed{I}\!- \; |\psi\rangle$$

In this circuit, applying the Identity gate to $|\psi\rangle$ leaves the qubit in state $|\psi\rangle$. This can be useful in maintaining dimensional consistency in quantum operations where certain qubits require no change.

Pauli-X Gate

The **Pauli-X gate**, also known as the quantum NOT gate, flips the state of a qubit. It maps the basis state $|0\rangle$ to $|1\rangle$ and $|1\rangle$ to $|0\rangle$. This gate is analogous to the classical NOT gate in classical computing.The Pauli-X gate has the following matrix form:

$$X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \tag{3.28}$$

When applied to a qubit in the state $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$, the Pauli-X gate produces the state $\alpha|1\rangle + \beta|0\rangle$.An example circuit using the Pauli-X gate can be illustrated as follows:

$$|0\rangle \; -\!\boxed{X}\!- \; |1\rangle$$

Pauli-Y Gate

The **Pauli-Y gate** is one of the three Pauli gates, along with the Pauli-X and Pauli-Z gates. The Pauli-Y gate is a single-qubit gate that performs a 180-degree rotation around the Y-axis on the Bloch sphere. This gate has significant applications in creating complex rotations and transformations, especially in conjunction with other Pauli and rotation gates.The Pauli-Y gate, often represented by Y, is defined by the following 2×2 matrix:

$$Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} \tag{3.29}$$

This matrix has complex off-diagonal elements, distinguishing it from the Pauli-X and Pauli-Z gates, which have real entries. The Pauli-Y gate acts as a reflection, incorporating a phase shift of $\pm i$, and can also be seen as a combination of the Pauli-X and Pauli-Z gates.The Pauli-Y gate acts on the computational basis states $|0\rangle$ and $|1\rangle$ as follows:

$$Y|0\rangle = i|1\rangle, \quad Y|1\rangle = -i|0\rangle \tag{3.30}$$

This transformation shows that the Pauli-Y gate swaps the states $|0\rangle$ and $|1\rangle$ with an additional phase factor of $\pm i$. This property is useful in applications that require phase manipulation or conditional operations, especially in quantum algorithms that need complex rotations.On the Bloch sphere, the Pauli-Y gate corresponds to a 180-degree rotation around the Y-axis. This rotation takes the state vector from the positive Z-axis to the negative Z-axis or vice versa. When applied to a qubit in the state $|+\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$, the Pauli-Y gate results in a state of $|-i\rangle = \frac{1}{\sqrt{2}}(|0\rangle - i|1\rangle)$, indicating the addition of a phase shift.The Pauli-Y gate is often depicted in quantum circuits with a Y symbol on a single qubit line. For a qubit initialized in the state $|\psi\rangle$, the Pauli-Y gate circuit is represented as follows:

$$|\psi\rangle \;-\!\boxed{Y}\!-$$

Here, applying the Pauli-Y gate on $|\psi\rangle$ introduces a 180-degree rotation around the Y-axis, effectively altering the qubit's phase. This gate is particularly useful in generating complex rotations and in scenarios where phase shifts are required, such as interference-based quantum algorithms.

Pauli-Z Gate

The **Pauli-Z gate** is one of the three Pauli gates, along with the Pauli-X and Pauli-Y gates. It is a single-qubit gate that performs a 180-degree rotation around the Z-axis on the Bloch sphere, effectively flipping the phase of a qubit without altering its amplitude. The Pauli-Z gate is often used in quantum algorithms for operations that require phase inversion and is essential in error correction and controlled operations.The Pauli-Z gate, denoted as Z, is represented by the following 2×2 matrix:

$$Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix} \tag{3.31}$$

This matrix has eigenvalues of $+1$ and -1, meaning it leaves the $|0\rangle$ state unchanged but flips the sign of the $|1\rangle$ state. As a result, the Pauli-Z gate applies a phase shift to states in the superposition of $|0\rangle$ and $|1\rangle$.The action of the Pauli-Z gate on the computational basis states is given by:

$$Z|0\rangle = |0\rangle, \quad Z|1\rangle = -|1\rangle \tag{3.32}$$

This transformation shows that the Pauli-Z gate leaves the $|0\rangle$ state unchanged but inverts the phase of the $|1\rangle$ state. When applied to a qubit in superposition, such as $|+\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$, the Pauli-Z gate creates a phase shift, resulting in $|-\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$.On the Bloch sphere, the Pauli-Z gate corresponds to a 180-degree rotation around the Z-axis. This rotation leaves the states on the Z-axis (such as $|0\rangle$ and $|1\rangle$) unaffected but inverts the phase of states in the X-Y plane. For example, applying the Pauli-Z gate to the state $|+\rangle$ shifts it to the state $|-\rangle$, illustrating the gate's role as a phase-flip operation.The Pauli-Z gate is also known as the phase-flip gate because it introduces a relative phase shift between the $|0\rangle$ and $|1\rangle$ states. This phase-flip property is crucial in quantum algorithms and error correction codes, where the Pauli-Z gate is often used in combination with other gates to create controlled phase shifts or to correct phase errors.In a quantum circuit, the Pauli-Z gate is represented by a Z symbol on a single qubit line. For a qubit initialized in the state $|\psi\rangle$, the circuit with the Pauli-Z gate applied is shown as:

$$|\psi\rangle \;\, —\boxed{Z}—$$

In this circuit, applying the Pauli-Z gate introduces a 180-degree phase shift if the qubit is in the $|1\rangle$ state, while leaving the $|0\rangle$ state unchanged. This operation is fundamental in constructing complex quantum algorithms that rely on precise control of qubit phases.

Hadamard Gate

The **Hadamard gate** is a fundamental quantum gate that creates superposition states. When applied to a qubit, it transforms the computational basis states $|0\rangle$ and $|1\rangle$ into an equal superposition of both states, effectively placing the qubit in a state where measurement outcomes for $|0\rangle$ and $|1\rangle$ are equally probable. This gate is crucial in many quantum algorithms, as it enables the simultaneous exploration of multiple states.The Hadamard gate, denoted by H, is represented by the following matrix:

$$H = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} \tag{3.33}$$

This matrix form illustrates that the Hadamard gate applies an equal-weighted transformation to both computational basis states, facilitating superposition by balancing the amplitude of each basis state.The Hadamard gate's effect on the basis states $|0\rangle$ and $|1\rangle$ is as follows:

$$H|0\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \tag{3.34}$$

$$H|1\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle) \tag{3.35}$$

Applying H to the state $|0\rangle$ results in an equal superposition state, where both $|0\rangle$ and $|1\rangle$ have an amplitude of $\frac{1}{\sqrt{2}}$. Similarly, applying H to $|1\rangle$ produces a state with a positive amplitude for $|0\rangle$ and a negative amplitude for $|1\rangle$, effectively creating a balanced but phase-inverted superposition.The circuit representation for applying the Hadamard gate to an initial qubit in the state $|0\rangle$ is shown below:

$$|0\rangle \; - \boxed{H} - \; \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$$

In this configuration, the qubit initially in state $|0\rangle$ is transformed by the Hadamard gate into a superposition state $\frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$, where both $|0\rangle$ and $|1\rangle$ are equally probable upon measurement.

Controlled-NOT (CNOT) Gate

The **Controlled-NOT (CNOT) gate** is a fundamental two-qubit gate in quantum computing. It performs a conditional operation where the state of the second qubit, called the target qubit, is flipped if and only if the first qubit, known as the control qubit, is in the state $|1\rangle$. The CNOT gate is essential in constructing entangled states, which are a key resource in quantum algorithms and quantum communication protocols.The CNOT gate can be represented by the following 4×4 unitary matrix, which operates on the two-qubit state vector:

$$\text{CNOT} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \tag{3.36}$$

This matrix demonstrates the conditional action on the target qubit based on the state of the control qubit. Specifically, when the control qubit is in the state $|1\rangle$, the CNOT gate swaps the target qubit's $|0\rangle$ and $|1\rangle$ states.The CNOT gate's action on the computational basis states is described below:

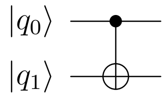
$$\text{CNOT}\,|00\rangle = |00\rangle \tag{3.37}$$

$$\text{CNOT}\,|01\rangle = |01\rangle \tag{3.38}$$

$$\text{CNOT}\,|10\rangle = |11\rangle \tag{3.39}$$

$$\text{CNOT}\,|11\rangle = |10\rangle \tag{3.40}$$

In this operation, when the control qubit is in the state $|0\rangle$, the target qubit remains unaffected. However, if the control qubit is in the state $|1\rangle$, the CNOT gate inverts the state of the target qubit. This conditional operation is integral to creating entanglement, as it establishes a correlation between the control and target qubits' states.The following circuit illustrates the application of the CNOT gate, where $|q_0\rangle$ serves as the control qubit and $|q_1\rangle$ as the target qubit:



In this example, the state of $|q_1\rangle$ is flipped if $|q_0\rangle$ is in the state $|1\rangle$. If $|q_0\rangle = |0\rangle$, the state of $|q_1\rangle$ remains unchanged. This control-target dynamic allows the CNOT gate to form entangled states, such as the Bell states, which are foundational in quantum information processing.

Multi-Controlled NOT (MCNOT) Gate

The **Multi-Controlled NOT (MCNOT) gate** is an extension of the basic Controlled-NOT (CNOT) gate. In the MCNOT gate, multiple qubits serve as control qubits, and the state of a single target qubit is flipped if and only if all control qubits are in the state $|1\rangle$. This gate is essential in complex quantum algorithms, as it allows for conditional operations based on multiple qubits' states, and is crucial in implementing functions such as multi-qubit conditional operations and encoding complex logic.The matrix representation of the MCNOT gate grows exponentially with the number of qubits involved, making it difficult to represent explicitly for large numbers of control qubits. However, the gate can be defined functionally: if all control qubits are in the state $|1\rangle$, the target qubit is flipped; otherwise, it remains unchanged. For example, a three-qubit MCNOT gate with two control qubits C_1 and C_2 and one target qubit T has the following action:

$$\text{MCNOT}\ |C_1C_2T\rangle = \begin{cases} |C_1C_2\overline{T}\rangle & \text{if } C_1 = 1 \text{ and } C_2 = 1 \\ |C_1C_2T\rangle & \text{otherwise} \end{cases} \tag{3.41}$$

where $\overline{T}$ represents the flipped state of the target qubit T.The effect of the MCNOT gate on computational basis states depends on the states of the control qubits. For an n-qubit MCNOT gate with $(n-1)$ control qubits and one target qubit, the action is as follows: i) If all control qubits are in the state $|1\rangle$, the target qubit is flipped from $|0\rangle$ to $|1\rangle$ or from $|1\rangle$ to $|0\rangle$. ii) If any control qubit is in the state $|0\rangle$, the target qubit remains unchanged.For example, with two control qubits and one target qubit, the action on the basis states is:

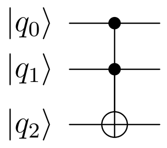
$$\text{MCNOT}\ |110\rangle = |111\rangle \tag{3.42}$$

$$\text{MCNOT}\ |111\rangle = |110\rangle \tag{3.43}$$

$$\text{MCNOT}\ |100\rangle = |100\rangle \tag{3.44}$$

$$\text{MCNOT}\ |101\rangle = |101\rangle \tag{3.45}$$

This property enables the MCNOT gate to implement conditional logic over multiple control qubits, an essential feature for many quantum algorithms.A circuit representation of the three-qubit MCNOT gate, where $|q_0\rangle$ and $|q_1\rangle$ are control qubits and $|q_2\rangle$ is the target qubit, is illustrated as follows:



In this circuit, the target qubit $|q_2\rangle$ is flipped only if both control qubits $|q_0\rangle$ and $|q_1\rangle$ are in the state $|1\rangle$. If either control qubit is in the state $|0\rangle$, $|q_2\rangle$ remains unchanged. This controlled operation can be scaled to larger numbers of control qubits, facilitating complex, multi-qubit conditional transformations.

The Controlled-Z (CZ) Gate

The **Controlled-Z (CZ) gate** is a fundamental two-qubit quantum gate widely used in quantum computing and quantum information processing. It performs a conditional operation that applies the Pauli-Z transformation to a target qubit depending on the state of a control qubit. Specifically, the CZ gate applies a phase flip to the target qubit when the control qubit is in the $|1\rangle$ state, leaving the qubits unchanged if the control qubit is in the $|0\rangle$ state.

Mathematically, the CZ gate is represented by a 4×4 unitary matrix in the computational basis

$\{|00\rangle, |01\rangle, |10\rangle, |11\rangle\}$:

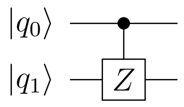
$$\text{CZ} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{pmatrix}. \tag{3.46}$$

In this matrix, the states $|00\rangle, |01\rangle$, and $|10\rangle$ are unchanged (multiplied by 1), while the state $|11\rangle$ undergoes a phase flip (multiplied by -1). This selective phase flip is essential in quantum algorithms where entanglement and conditional operations are needed to create interference patterns for correct computation outcomes.

The action of the CZ gate can be better understood by examining how it operates on a two-qubit state $|\psi\rangle = \alpha|00\rangle + \beta|01\rangle + \gamma|10\rangle + \delta|11\rangle$, where $\alpha, \beta, \gamma,$ and δ are complex amplitudes. Applying the CZ gate yields:

$$\text{CZ}|\psi\rangle = \alpha|00\rangle + \beta|01\rangle + \gamma|10\rangle - \delta|11\rangle \tag{3.47}$$

Here, the amplitude associated with the $|11\rangle$ component is negated, which is equivalent to applying a Pauli-Z gate on the target qubit only when the control qubit is in the $|1\rangle$ state.In quantum circuit diagrams, the CZ gate is typically represented with a control dot on the control qubit and a Z-gate on the target qubit, connected by a vertical line. For example, in the following circuit, qubit $|q_0\rangle$ serves as the control qubit, while qubit $|q_1\rangle$ is the target qubit:



The CZ gate plays a crucial role in creating entanglement, which is a cornerstone of quantum computing. For example:

- Bell State Generation: By combining a Hadamard gate on the control qubit with a CZ gate, we can generate maximally entangled Bell states, such as $\frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$.

- Quantum Algorithms: In algorithms like Grover's search and the Quantum Fourier Transform, CZ gates enable controlled operations that entangle qubits, allowing for interference patterns that yield correct outputs.

- Error Correction and Stabilizers: In quantum error correction, CZ gates are used in stabilizer codes to detect and correct errors across entangled qubits.

The controlled-Z gate is thus indispensable for quantum operations requiring entanglement and conditional phase flips, enabling complex quantum states that are essential in advanced quantum algorithms and protocols.

SWAP Gate

The **SWAP gate** is a two-qubit gate that interchanges the states of two qubits. Unlike the CNOT or MCNOT gates, the SWAP gate does not involve conditional operations but rather performs a direct exchange of the qubits' states. It is an essential gate in quantum circuits, especially when qubits need to be relocated or reordered within a circuit. This gate can also be decomposed into a series of CNOT gates, enabling efficient implementation in quantum hardware.The SWAP gate is represented by a 4×4 unitary matrix, which acts on the two-qubit state vector:

$$\text{SWAP} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{3.48}$$

In this matrix, the off-diagonal entries indicate that the states of the two qubits are exchanged. Specifically, $|01\rangle$ and $|10\rangle$ are swapped, while the states $|00\rangle$ and $|11\rangle$ remain unchanged.The SWAP gate acts on the computational basis states by exchanging the states of the two qubits as follows:

$$\text{SWAP}\,|00\rangle = |00\rangle \tag{3.49}$$

$$\text{SWAP}\,|01\rangle = |10\rangle \tag{3.50}$$

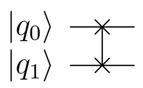
$$\text{SWAP}\,|10\rangle = |01\rangle \tag{3.51}$$

$$\text{SWAP}\,|11\rangle = |11\rangle \tag{3.52}$$

This action demonstrates that the SWAP gate leaves the states $|00\rangle$ and $|11\rangle$ unchanged but interchanges $|01\rangle$ and $|10\rangle$. This property is particularly useful for quantum circuits that require qubit reordering, such as certain quantum algorithms or when specific qubits need to interact based on physical qubit layout constraints in quantum hardware.The SWAP gate can be decomposed into a sequence of three CNOT gates, enabling practical implementation on quantum devices that may not support SWAP gates directly. The decomposition is as follows:

$$\text{SWAP} = \text{CNOT}_{1,2}\text{CNOT}_{2,1}\text{CNOT}_{1,2} \tag{3.53}$$

This sequence of operations achieves the same effect as a single SWAP gate by conditionally flipping the target qubits based on the states of the control qubits.The following quantum circuit illustrates the SWAP gate applied to two qubits $|q_0\rangle$ and $|q_1\rangle$:



In this circuit, the states of $|q_0\rangle$ and $|q_1\rangle$ are interchanged. This operation is equivalent to transferring the information in $|q_0\rangle$ to $|q_1\rangle$ and vice versa, allowing qubit reordering without disturbing the overall quantum state. The SWAP gate is thus highly valuable for optimizing quantum circuit layout and managing qubit connectivity constraints.

Controlled-SWAP Gate

The **Controlled-SWAP gate** is a three-qubit gate that conditionally swaps the states of two qubits based on the state of a third qubit, called the control qubit. If the control qubit is in the state $|1\rangle$, the states of the two target qubits are swapped; if the control qubit is in the state $|0\rangle$, the target qubits remain unchanged. The controlled-SWAP gate is valuable in reversible computing and is widely used in quantum circuits that require conditional reordering of qubit states.The controlled-SWAP gate can be represented by an 8×8 matrix, acting on the three-qubit state vector. This matrix swaps the last two qubits if the control qubit (first qubit) is in the state $|1\rangle$, while leaving the state unchanged if the control qubit is in the state $|0\rangle$. The matrix form is:

$$\text{CSWAP} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix} \tag{3.54}$$

Here, the entries reflect that the controlled-SWAP gate conditionally exchanges the states of the last two qubits when the control qubit is in state $|1\rangle$.The action of the controlled-SWAP gate on computational basis states is as follows:

$$\text{CSWAP}\,|000\rangle = |000\rangle \tag{3.55}$$

$$\text{CSWAP}\,|001\rangle = |001\rangle \tag{3.56}$$

$$\text{CSWAP}\,|010\rangle = |010\rangle \tag{3.57}$$

$$\text{CSWAP}\,|011\rangle = |011\rangle \tag{3.58}$$

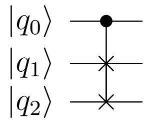
$$\text{CSWAP}\,|100\rangle = |100\rangle \tag{3.59}$$

$$\text{CSWAP}\,|101\rangle = |110\rangle \tag{3.60}$$

$$\text{CSWAP}\,|110\rangle = |101\rangle \tag{3.61}$$

$$\text{CSWAP}\,|111\rangle = |111\rangle \tag{3.62}$$

In these transformations, the first qubit is the control, while the last two qubits are the targets. The states $|101\rangle$ and $|110\rangle$ are swapped when the control qubit is $|1\rangle$, illustrating the gate's conditional operation.The controlled-SWAP gate can be decomposed into a series of elementary gates, such as CNOT and Toffoli (CCNOT) gates, to simplify its implementation on quantum hardware. The decomposition is typically: 1. Apply a CNOT gate between one of the target qubits and the other target qubit, controlled by the first target. 2. Use a Toffoli gate with the control qubit and the first target to achieve the conditional swapping. This decomposition allows the controlled-SWAP gate to be realized on hardware platforms that natively support only single- and two-qubit operations.The following circuit shows the controlled-SWAP gate acting on three qubits: $|q_0\rangle$ as the control qubit and $|q_1\rangle$, $|q_2\rangle$ as the target qubits:



In this circuit, $|q_1\rangle$ and $|q_2\rangle$ are swapped if $|q_0\rangle = |1\rangle$; otherwise, they remain in their initial states. This operation is instrumental in conditional operations where qubit states need to be conditionally reordered.

3.10 Quantum Measurement

Quantum measurement is a core concept in quantum mechanics. The process of observation causes a quantum system to collapse into a definite state. This differs from classical measurement, which simply reveals pre-existing properties. Instead, quantum measurement significantly alters the measured system, making it a unique and complex operation [1]. To simply say, it involves observing a quantum system, which causes its state to collapse into one of the eigenstates of the observable being measured. Here's a mathematical intuition using an example:

Example: Measurement of a Qubit State

Suppose a qubit is in a superposition state:

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle \tag{3.63}$$

where $|0\rangle$ and $|1\rangle$ are the basis states (eigenstates of the Z-observable), and $\alpha, \beta \in \mathbb{C}$ with $|\alpha|^2 + |\beta|^2 = 1$.

The measurement operator corresponds to an observable Z, which has eigenstates $|0\rangle$ and $|1\rangle$ with eigenvalues $+1$ and -1, respectively. Mathematically:

$$Z = |0\rangle\langle 0| - |1\rangle\langle 1| \tag{3.64}$$

When we measure Z, the quantum state collapses to one of its eigenstates. The probabilities of obtaining these outcomes are:

$$P(0) = |\langle 0|\psi\rangle|^2 = |\alpha|^2 \tag{3.65}$$

$$P(1) = |\langle 1|\psi\rangle|^2 = |\beta|^2 \tag{3.66}$$

The post measurement state is given by:

- If the measurement result is $+1$, the state collapses to $|0\rangle$, and the system is now in the state:
$$|\psi_{\text{post}}\rangle = |0\rangle \tag{3.67}$$
- If the measurement result is -1, the state collapses to $|1\rangle$, and the system is now in the

state:

$$|\psi_{\text{post}}\rangle = |1\rangle \tag{3.68}$$

Figure 3.6, represents the difference between the classical and the quantum measurement.

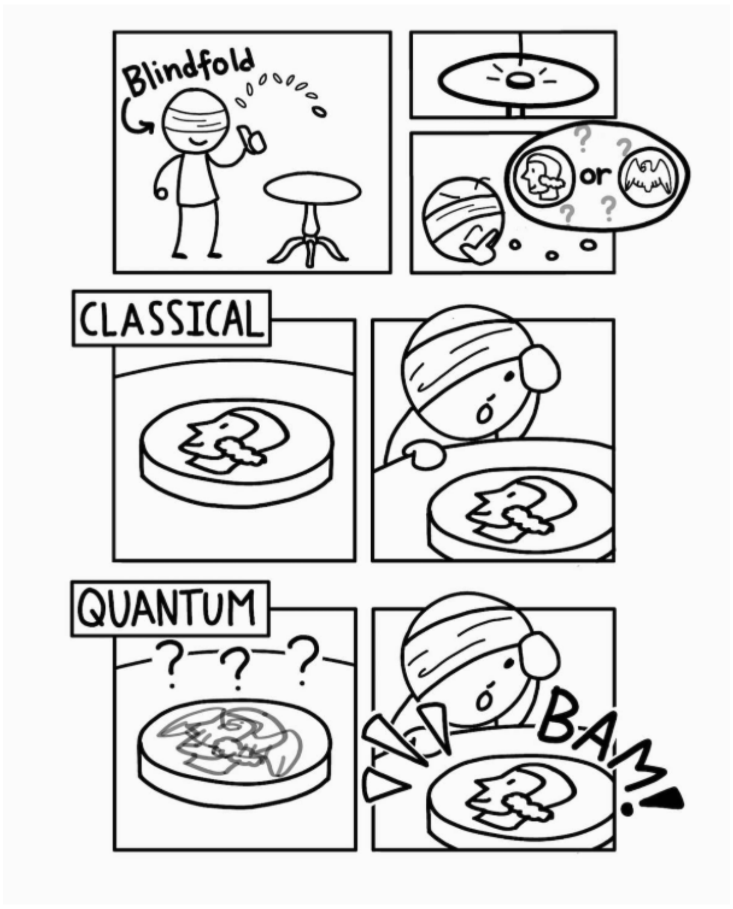


Figure 3.6: Comparison between classical and quantum measurement outcomes [1, 2].

The mathematical principles underlying quantum measurement elucidate its probabilistic essence, which is pivotal for comprehending quantum computing algorithms and physical realizations [42, 60].

3.11 The No-Cloning Theorem

The **No-Cloning Theorem** is a fundamental result in quantum mechanics, claiming that it is impossible to produce a perfect clone of any unknown quantum state [61]. This theorem is important in quantum information theory because it ensures the security of quantum communication protocols like Quantum Key Distribution (QKD).Let $|\psi\rangle$ and $|\phi\rangle$ be arbitrary quantum states in a Hilbert space. If a hypothetical unitary operator U exists that performs cloning:

$$U(|\psi\rangle \otimes |x\rangle) = |\psi\rangle \otimes |\psi\rangle \tag{3.69}$$

for all states $|\psi\rangle$, then U must also satisfy:

$$U\left(|\phi\rangle \otimes |x\rangle\right) = |\phi\rangle \otimes |\phi\rangle \tag{3.70}$$

However, such a unitary transformation U cannot exist for arbitrary $|\psi\rangle$ and $|\phi\rangle$.This finding is intimately related to the linearity of quantum physics and the features of Hilbert spaces. It emphasizes an important distinction between classical and quantum information, assuring the inherent security of quantum communication protocols.

3.12 Quantum Algorithms

Quantum computing is a revolutionary field that leverages the principles of quantum mechanics to solve problems exponentially faster than classical computers. Classical algorithms process information using bits (0 or 1), whereas quantum algorithms utilize qubits, which can exist in a superposition of both 0 and 1 simultaneously. Additionally, quantum entanglement allows qubits to share information in ways impossible for classical bits.

This chapter introduces three foundational quantum algorithms:

- **Deutsch-Jozsa Algorithm** – Determines whether a function is constant or balanced with just one query.
- **Grover's Algorithm** – Finds a marked item in an unsorted dataset with quadratic speedup over classical searching.
- **Shor's Algorithm** – Factorizes large integers exponentially faster than classical algorithms.
- **Quantum Approximate Optimization Algorithm (QAOA)** – Solves combinatorial optimization problems efficiently using quantum-classical hybrid optimization.

Each algorithm is explained in an easy-to-understand manner, including its problem statement, mathematical formulation, and quantum circuit representation using diagrams.

3.12.1 Deutsch-Jozsa Algorithm

Problem Statement

The Deutsch-Jozsa algorithm is a quantum algorithm designed to determine whether a given Boolean function $f : \{0,1\}^n \to \{0,1\}$ is constant (same output for all inputs) or balanced (half the inputs yield 0, half yield 1) in a single query. Classically, this would require $2^{n-1}+1$ queries in the worst case.

Mathematical Representation

1. Initialize the qubits: We start with n qubits in the $|0\rangle$ state and one auxiliary qubit in $|1\rangle$:

$$|\psi_0\rangle = |0\rangle^{\otimes n} \otimes |1\rangle \tag{3.71}$$

2. Apply Hadamard gates to create a superposition of all possible inputs:

$$|\psi_1\rangle = H^{\otimes(n+1)}|\psi_0\rangle \tag{3.72}$$

3. Query the function using an oracle U_f:

$$U_f|x\rangle|y\rangle = |x\rangle|y \oplus f(x)\rangle \tag{3.73}$$

4. Apply Hadamard gates again to the first n qubits and measure the result. If the output is $|0\rangle^{\otimes n}$, the function is constant; otherwise, it is balanced.

Algorithm 1 Deutsch-Jozsa Algorithm

```
Require: Oracle function $f : \{0,1\}^n \to \{0,1\}$
Ensure: Determine if $f(x)$ is constant or balanced
 1: Initialize $(n+1)$ qubits in the state:
 2:         $|\psi_0\rangle = |0\rangle^{\otimes n} |1\rangle$
 3: Apply Hadamard transform on all qubits:
 4:         $|\psi_1\rangle = H^{\otimes(n+1)} |\psi_0\rangle$
 5: Apply the Oracle function $U_f$:
 6:         $|x\rangle |y\rangle \to |x\rangle |y \oplus f(x)\rangle$
 7: Apply Hadamard transform on the first $n$ qubits:
 8:         $|\psi_2\rangle = H^{\otimes n} |x\rangle |y\rangle$
 9: Measure the first $n$ qubits.
10: if Measurement result is $|0\rangle^{\otimes n}$ then
11:     return "Function is constant"
12: else
13:     return "Function is balanced"
14: end if
```

Circuit Representation

Figure 3.7 represents the circuit needed to implement the Deutsch–Jozsa Algorithm.

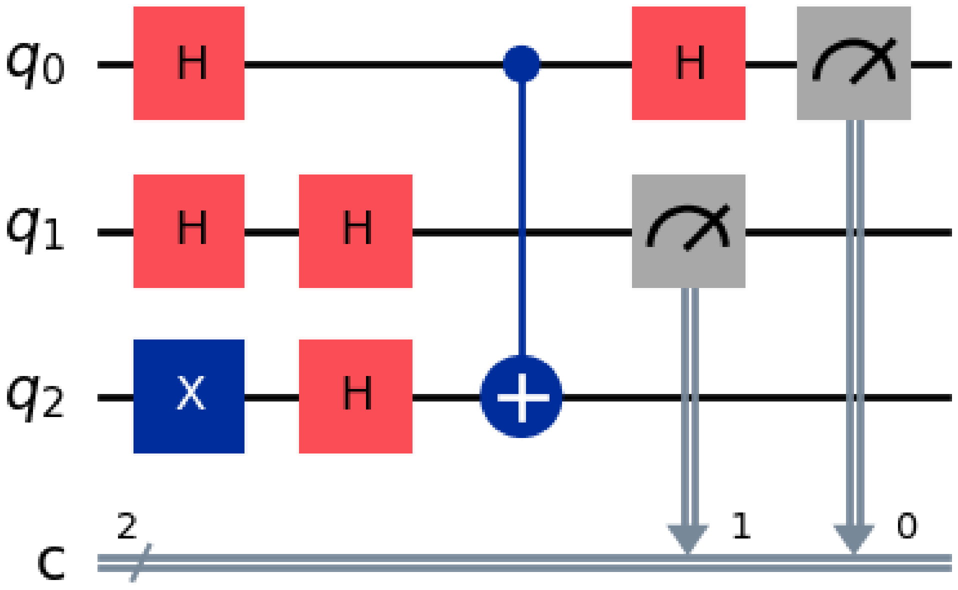


Figure 3.7: Quantum Circuit Diagram for the Deutsch–Jozsa Algorithm, illustrating the initialization, oracle (balanced or constant), and interference via Hadamard transforms.

3.12.2 Grover's Algorithm

Problem Statement

Grover's algorithm is a quantum search algorithm that finds a marked item in an unsorted database of $N = 2^n$ elements in $O(\sqrt{N})$ time instead of $O(N)$ classically. This offers a quadratic speedup over classical brute-force search.

Mathematical Representation

1. Initialize all qubits in a superposition state:

$$|\psi_0\rangle = H^{\otimes n}|0\rangle^{\otimes n} \tag{3.74}$$

2. Apply the Oracle operator U_f to mark the correct solution:

$$U_f|x\rangle = (-1)^{f(x)}|x\rangle \tag{3.75}$$

3. Apply the Diffusion Operator to amplify the probability of the correct answer:

$$U_s = 2|s\rangle\langle s| - I \tag{3.76}$$

4. Repeat steps 2 and 3 approximately $O(\sqrt{N})$ times.

5. Measure the state to obtain the solution.

Algorithm 2 Grover's Algorithm for Unstructured Search

```
Require: Oracle function $O_f$, number of qubits $n$, number of iterations $T = \lfloor \frac{\pi}{4}\sqrt{2^n} \rfloor$
Ensure: Index $x^*$ such that $f(x^*) = 1$
1: Initialize $n$ qubits in uniform superposition:
2:         $|\psi\rangle = H^{\otimes n} |0\rangle^{\otimes n}$
3: for $t = 1$ to $T$ do
4:     Apply Oracle $O_f$: $|x\rangle \to (-1)^{f(x)} |x\rangle$
5:     Apply Grover Diffusion Operator:
6:             $D = 2 |\psi\rangle \langle\psi| - I$
7: end for
8: Measure and obtain $x^*$.
9: return $x^*$
```

Circuit Representation

Figure 3.8 represents the circuit needed to implement the Grover's Algorithm.

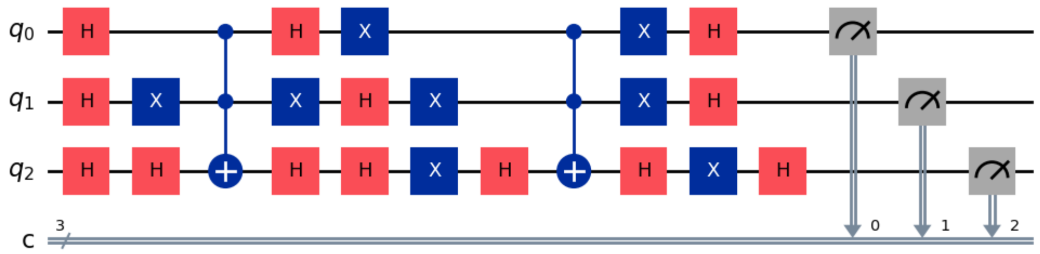


Figure 3.8: Quantum Circuit Diagram for Grover's Algorithm: Oracle and Diffuser Implementation for Marking $|101\rangle$

3.12.3 Shor's Algorithm

Problem Statement

Shor's algorithm efficiently factorizes large integers in polynomial time, a task that classical computers solve exponentially slowly. It relies on quantum period-finding using the Quantum Fourier Transform (QFT).

Mathematical Representation

1. Choose a random number a and define the modular function:

$$f(x) = a^x \mod N \tag{3.77}$$

2. Prepare a quantum superposition over all possible values of x.

3. Apply the Quantum Fourier Transform (QFT) to extract the period r:

$$|\psi_2\rangle = QFT|\psi_1\rangle \tag{3.78}$$

4. Measure the output and use continued fractions to determine r.

5. Compute the greatest common divisor:

$$\gcd(a^{r/2} - 1, N) \tag{3.79}$$

to find a non-trivial factor of N.

Algorithm 3 Shor's Algorithm for Integer Factorization

```
Require: Composite number $N = p \cdot q$ (odd, not prime), choose random $a < N$
Ensure: Non-trivial factors of $N$
1: Compute $\gcd(a, N)$. If $\gcd(a, N) \neq 1$, return it as a factor.
2: Use quantum phase estimation to find the order $r$ of $a$ modulo $N$:
3:         $a^r \equiv 1 \pmod{N}$
4: if $r$ is odd or $a^{r/2} \equiv -1 \pmod{N}$ then
5:     Restart with a new random $a$.
6: end if
7: Compute $\gcd(a^{r/2} - 1, N)$ and $\gcd(a^{r/2} + 1, N)$.
8: return Non-trivial factors of $N$.
```

Circuit Representation

Figure 3.9, represents the circuit needed to implement Shor's Algorithm.

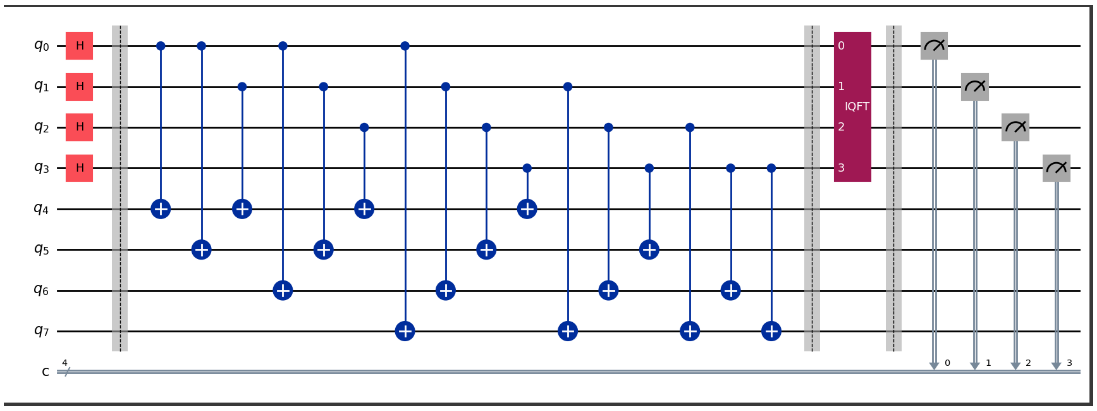


Figure 3.9: Quantum Circuit Diagram for Shor's Algorithm (Period-Finding Subroutine)

3.12.4 Quantum Approximate Optimization Algorithm (QAOA)

The Quantum Approximate Optimization Algorithm (QAOA) is a variational quantum algorithm designed to solve combinatorial optimization problems. It employs a hybrid quantum-classical approach, leveraging quantum mechanics to approximate optimal solutions efficiently.

Problem Formulation

Given a cost function $C(x)$ over binary strings $x \in \{0,1\}^n$, the optimization problem is formulated as finding:

$$x^* = \arg\min_x C(x). \tag{3.80}$$

The corresponding quantum cost Hamiltonian is:

$$H_C = \sum_x C(x) \left| x \right\rangle \left\langle x \right| , \tag{3.81}$$

where the ground state of H_C encodes the optimal solution.

QAOA Ansatz

QAOA alternates between the cost Hamiltonian H_C and a mixing Hamiltonian H_B, where:

$$H_B = \sum_i X_i, \tag{3.82}$$

with X_i being the Pauli-X operator on qubit i. The quantum state is evolved as:

$$|\psi(\gamma,\beta)\rangle = U(H_B,\beta_p)U(H_C,\gamma_p)\dots U(H_B,\beta_1)U(H_C,\gamma_1)|\psi_0\rangle, \tag{3.83}$$

where the unitary operators are defined as:

$$U(H_C,\gamma) = e^{-i\gamma H_C}, \quad U(H_B,\beta) = e^{-i\beta H_B}. \tag{3.84}$$

The initial state is prepared as:

$$|\psi_0\rangle = H^{\otimes n}|0\rangle^{\otimes n}. \tag{3.85}$$

Optimization Process

The objective function to minimize is:

$$F(\gamma,\beta) = \langle\psi(\gamma,\beta)|H_C|\psi(\gamma,\beta)\rangle. \tag{3.86}$$

A classical optimization routine updates the parameters (γ,β) iteratively to find the optimal values that minimize $F(\gamma,\beta)$.

Algorithm 4 Quantum Approximate Optimization Algorithm (QAOA)

```
Require: Objective function $C$, number of qubits $n$, number of iterations $p$, initial parameters $(\boldsymbol{\beta},\boldsymbol{\gamma})$
Ensure: Optimized parameters $(\boldsymbol{\beta}^*,\boldsymbol{\gamma}^*)$ and approximate solution to $C$
 1: Initialize qubits in a uniform superposition state:
 2:         $|\psi_0\rangle = H^{\otimes n}|0\rangle^{\otimes n}$
 3: for $k = 1$ to $p$ do
 4:     Apply the cost Hamiltonian:
 5:             $|\psi_k\rangle = e^{-i\gamma_k H_C}|\psi_{k-1}\rangle$
 6:     Apply the mixing Hamiltonian:
 7:             $|\psi_k\rangle = e^{-i\beta_k H_B}|\psi_k\rangle$
 8: end for
 9: Measure the quantum state $|\psi_p\rangle$ in the computational basis.
10: Compute the expectation value $\langle C\rangle$ based on the measurement outcomes.
11: Use a classical optimizer to update $(\boldsymbol{\beta},\boldsymbol{\gamma})$.
12: Repeat until convergence criteria is met.
13: return Optimized parameters $(\boldsymbol{\beta}^*,\boldsymbol{\gamma}^*)$ and approximate solution to $C$
```

QAOA provides a near-term quantum advantage for solving combinatorial optimization problems, making it a fundamental algorithm in quantum computing research.

Circuit Representation

Figure 3.10 represents the circuit needed to implement the Quantum Approximate Optmization Algorithm.

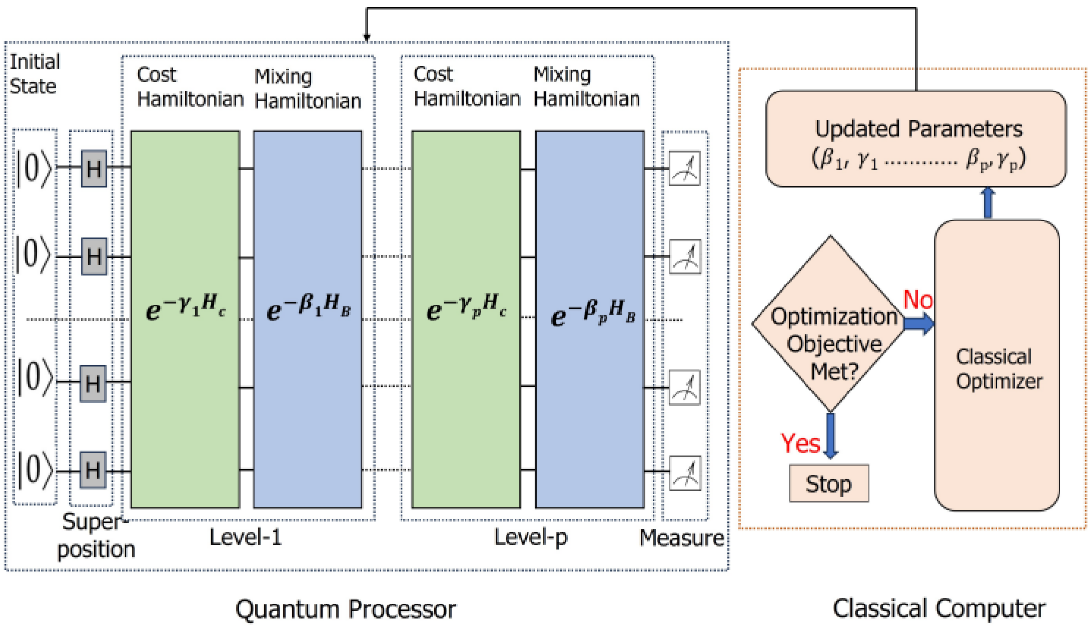


Figure 3.10: Quantum Approximate Optimization Algorithm (QAOA) workflow: The quantum processor applies cost and mixing Hamiltonians, while a classical optimizer updates parameters iteratively

Quantum algorithms such as Deutsch-Jozsa, Grover's, and Shor's algorithms illustrate the power of quantum computation. While Deutsch-Jozsa provides an exponential speedup, Grover's and Shor's algorithms demonstrate practical quantum advantages in search and cryptography.Quantum Approximate Optimization Algorithm (QAOA) has gained attention for its ability to tackle combinatorial optimization problems, offering a promising approach for real-world applications in logistics, finance, and network optimization. As quantum hardware evolves, these algorithms will play a crucial role in solving complex computational problems.

3.13 Application of Quantum Computing

Quantum computing has emerged as a transformative paradigm, harnessing the principles of quantum mechanics to tackle problems that are computationally intractable for classical systems. As this revolutionary technology progresses, it is being increasingly applied across a diverse array of fields, such as cryptography, optimization, machine learning, drug discovery, and materials science. This section will explore some of the key applications of quantum computing, highlighting its potential to address complex problems that were previously considered beyond the reach of classical computing approaches.

3.13.1 Quantum Cryptography

Quantum cryptography leverages the principles of quantum mechanics to bolster data security, providing provably secure communication protocols. One prominent example is Quantum Key Distribution, which employs quantum states to facilitate the exchange of cryptographic keys. Unlike classical approaches, any attempt to eavesdrop on the quantum system would disrupt it, alerting both parties to the breach in security [34] [32].

3.13.2 Quantum Machine Learning

Quantum computing and machine learning techniques are synergistically combined to accelerate model training and enhance predictive accuracy in Quantum Machine Learning. Algorithms such as Quantum Support Vector Machines and Quantum Neural Networks demonstrate potential in addressing high-dimensional data classification and clustering challenges more efficiently compared to classical approaches [62].

3.13.3 Quantum Optimization

Quantum computing shows promise in revolutionizing optimization problems, especially in areas like logistics, finance, and manufacturing. The Quantum Approximate Optimization Algorithm and Quantum Annealing have been proposed as methods to solve combinatorial optimization problems that involve searching for optimal solutions among numerous possibilities. These quantum approaches could have a transformative impact on domains such as supply chain management, drug discovery, and financial portfolio optimization [63].

3.13.4 Drug Discovery and Quantum Chemistry

Quantum computing has immense potential in the domains of quantum chemistry and drug discovery. Classical simulations of molecular systems are computationally intensive, particularly for large molecules with numerous interacting particles. Quantum computers can model quantum systems in a more natural and efficient manner, facilitating breakthroughs in comprehending complex chemical reactions and protein folding, thereby expediting the development of novel pharmaceuticals [64] [65].

3.13.5 Materials Science and Simulation

Quantum computing holds immense potential to bolster progress in materials science by facilitating the simulation of materials at the fundamental quantum level. Quantum-powered devices can model the electronic structures of materials, forecast the emergence of novel materials with targeted properties, and simulate the conduct of intricate systems that prove challenging or infeasible to study using classical computing approaches. This capability can catalyze breakthroughs in domains such as superconductivity, energy storage, and other materials-driven innovations [66].

3.13.6 Artificial Intelligence and Data Analysis

Quantum computing can complement artificial intelligence by expediting the processing of extensive data sets and facilitating more efficient decision-making. Quantum-inspired algorithms, such as Quantum Principal Component Analysis and Quantum Fourier Transform, have the potential to augment classical AI models, empowering them to recognize patterns and derive data-driven insights more effectively.

3.13.7 Quantum Networking and Communication

Quantum communication represents another crucial application of quantum computing, enabling the secure and high-fidelity transmission of information. Quantum networks leverage the principle of quantum entanglement to establish communication channels that are fundamentally secure, as any eavesdropping would disrupt the quantum state and be detectable by the parties involved. This capability is essential for the development of the future quantum internet infrastructure.

.

Chapter 4

Development of a Quantum K-Nearest Neighbors Classifier: From Theory to Implementation

4.1 Introduction

The proposed Quantum K-Nearest Neighbors algorithm presents a novel quantum-based approach to address the limitations of classical KNN in terms of computational efficiency and scalability for high-dimensional data. By leveraging the principles of quantum computation, such as quantum state encoding, quantum gate operations, and similarity measurement through swap tests, QKNN aims to enhance the performance of classification tasks.

The key steps of the QKNN methodology involve: i) encoding classical data into normalized quantum states, ii) applying quantum gates to encode features and relationships, and iii) performing swap tests to compute overlaps between quantum states. The resulting similarity scores are then used to determine the nearest neighbors, followed by a majority voting mechanism for the final classification.

A notable advantage of QKNN is the quantum hardware's ability to compute inner products between high-dimensional data points in a fundamentally different way than classical systems, potentially leading to faster similarity computations. By encoding both training and test data into quantum states, the QKNN algorithm achieves a form of quantum-enhanced parallelism, making it well-suited for applications involving large and complex datasets.

Furthermore, the QKNN methodology integrates hyperparameter tuning, such as setting similarity thresholds, to adapt to specific datasets. The algorithm is also designed to be hardware-efficient, considering the constraints of current quantum devices while remaining extensible to future advancements in quantum technology.

In summary, the QKNN algorithm introduces a quantum-enhanced framework for classification, addressing the computational bottlenecks of classical KNN and paving the way for more scalable and efficient machine learning methodologies in quantum computing. This chapter provides a detailed exploration of the proposed methodology, outlining its theoretical underpinnings, algorithmic steps, and potential for practical applications.

4.2 Problem Defination

The Quantum K-Nearest Neighbors algorithm addresses the challenge of efficiently classifying and analyzing high-dimensional datasets using quantum computing. Traditional K-Nearest Neighbors is a widely used supervised learning algorithm due to its simplicity and effectiveness. However, as dataset sizes and dimensionalities grow, classical KNN faces limitations in computational efficiency and scalability. Its performance can also degrade when dealing with complex, noisy, or overlapping data distributions.

QKNN overcomes these limitations by leveraging the inherent parallelism and speedup capabilities of quantum computation. The algorithm optimally handles high-dimensional data by performing distance computations and neighbor identification in the quantum domain, reducing computational overhead and improving classification performance. Furthermore, QKNN is well-suited for processing hybrid classical-quantum data, enabling its application to both classical datasets and quantum-native information. This allows QKNN to take advantage of the unique properties of quantum systems, such as superposition and entanglement, to achieve more efficient and accurate classification results compared to classical KNN approaches.

The QKNN algorithm represents a significant advancement in the field of quantum machine learning, providing a quantum-enhanced framework for addressing the computational challenges faced by classical KNN. By leveraging the parallelism and speed of quantum computation, QKNN has the potential to unlock new possibilities in high-dimensional data analysis and classification, paving the way for more scalable and efficient machine learning methodologies in quantum computing.

To validate its effectiveness, QKNN was tested on three real-life datasets: Breast Cancer, Iris, and Stroke Prediction. These datasets encompass a range of classification complexities and real-world relevance:

The **Iris dataset**, introduced by Fisher [67], consists of 150 samples from three species of iris flowers: *Iris setosa, Iris versicolor, and Iris virginica*. Each sample has four numerical features: sepal length, sepal width, petal length, and petal width, measured in centimeters. It is widely used in machine learning classification tasks.

The **Breast Cancer Wisconsin dataset**, provided by Wolberg [68], is used for binary classification of malignant and benign tumors. It contains 699 samples with 10 numerical features related to cell nucleus characteristics, aiding in breast cancer diagnosis.

The **Banknote Authentication dataset**, collected by Abbas [69], consists of 1372 samples with four numerical features extracted from genuine and counterfeit banknotes using wavelet analysis. It is used to classify banknotes as real or fake based on image features.

By applying QKNN to these datasets, this research demonstrates the algorithm's practical utility in handling real-world classification problems, showcasing its advantages in accuracy and computational efficiency over classical counterparts.

4.3 Theoretical Foundation

4.3.1 Classical K-Nearest Neighbors

The K-Nearest Neighbors algorithm is a simple yet powerful supervised machine learning method. It determines the class of a test instance based on the classes of its k nearest neighbors in the feature space. KNN is a non-parametric algorithm, meaning it does not make assumptions about the underlying data distribution. This grants it considerable flexibility. The core idea is that similar instances are likely to belong to the same class. The algorithm operates by storing all available training instances and classifying new instances based on a similarity measure, such as Euclidean distance, which quantifies the closeness between the feature vectors of the test instance and the training instances. Then K nearest neighbors are identified, and the test point is assigned the majority class label among these neighbors. Mathematically, for a test point x_{test}, the predicted class y_{pred} is:

$$y_{\text{pred}} = \arg\max_{c} \sum_{i=1}^{k} \mathbb{I}(y^{(i)} = c), \tag{4.1}$$

where $y^{(i)}$ is the class of the i-th neighbor, c is a candidate class, and $\mathbb{I}(\cdot)$ is the indicator function.Figure 4.1 represents the methodology of Classical KNN algorithm.

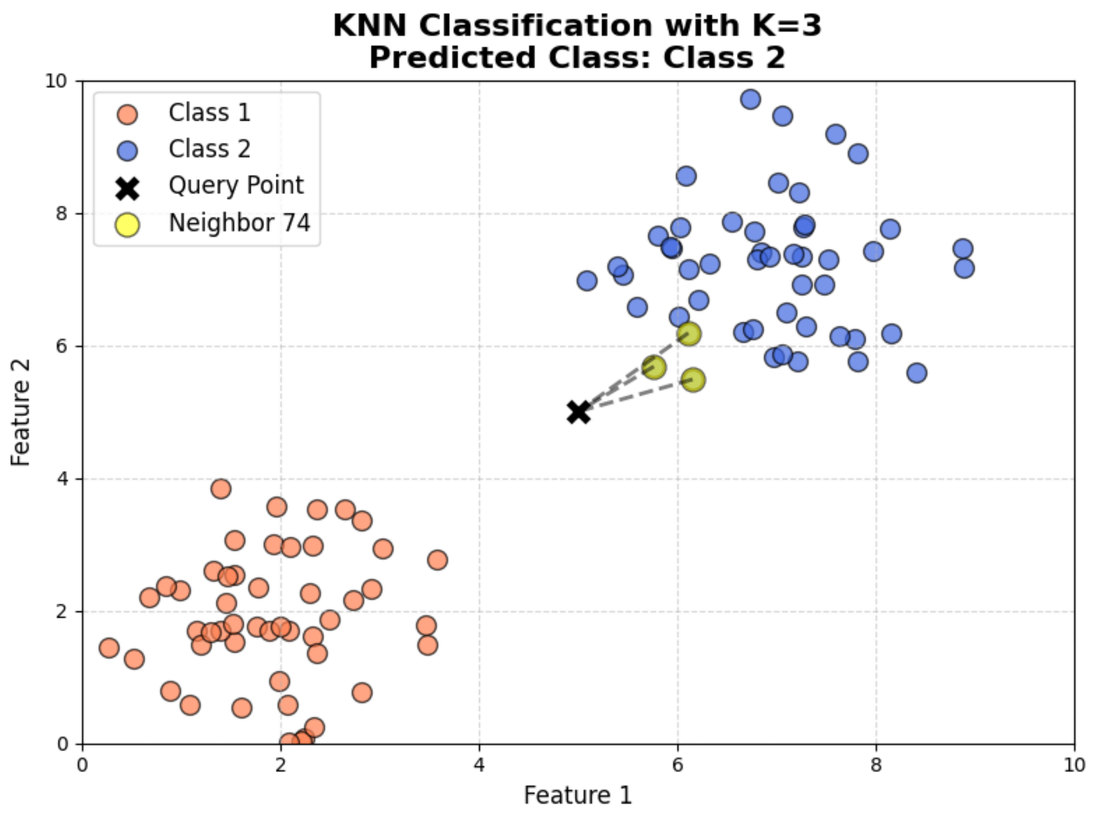


Figure 4.1: KNN Methodology

To compute the distance between the test data point and train data points, the choice of distance metric is critical in KNN as it determines the similarity between data points. Common distance metrics include:

Euclidean Distance:

$$d(\mathbf{x}_1, \mathbf{x}_2) = \sqrt{\sum_{j=1}^{n} (x_{1,j} - x_{2,j})^2} \tag{4.2}$$

Manhattan Distance:

$$d(\mathbf{x}_1, \mathbf{x}_2) = \sum_{j=1}^{n} |x_{1,j} - x_{2,j}| \tag{4.3}$$

Cosine Similarity (used for directional similarity rather than magnitude):

$$\text{Sim}(\mathbf{x}_1, \mathbf{x}_2) = \frac{\mathbf{x}_1 \cdot \mathbf{x}_2}{||\mathbf{x}_1|| ||\mathbf{x}_2||} \tag{4.4}$$

where $\mathbf{x}_1$ and $\mathbf{x}_2$ are data points, and n is the number of features. The metric is selected based on the nature of the data and the problem requirements.

The algorithm then identifies the k nearest neighbors to the test point based on their computed distances. The predicted class is then determined by the majority class label among these neighbors.

Algorithm 5 Classical k-NN Algorithm

```
Require: Training dataset $\mathcal{D} = \{(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)\}$, query instance $x_q$, number of
    neighbors $k$.
Ensure: Predicted class label $y_q$.
1: Compute distances $d(x_q, x_i)$ for all $i \in \{1, 2, \dots, n\}$.
2: Select the $k$ nearest neighbors $N_k$ based on the smallest distances.
3: Determine the predicted class:
4:         $y_q = \arg\max_{c \in C} \sum_{i \in N_k} I(y_i = c)$.
5: Return $y_q$.
```

While KNN is intuitive and straightforward to implement, it faces several limitations, particularly when applied to large-scale or high-dimensional datasets:

High Computational Complexity: The KNN algorithm requires calculating distances between the test instance and all training instances, resulting in a time complexity of $O(nd)$, where n is the number of training instances and d is the dimensionality of the feature space. This computation becomes increasingly resource-intensive as the dataset size grows.

Memory Constraints: KNN is a memory-intensive method that stores the entire training dataset in memory. This requirement makes it inefficient for applications involving large datasets or limited hardware resources.

Curse of Dimensionality: In high-dimensional feature spaces, the distance between data points becomes less meaningful, a phenomenon known as the curse of dimensionality. This can reduce the algorithm's accuracy and increase the computational burden due to the exponential growth of the search space.

The limitations of the classical KNN algorithm motivate the exploration of more advanced and efficient approaches, such as quantum-enhanced variants, to address the computational inefficiencies and scalability challenges inherent in the traditional method.

4.3.2 Quantum Machine Learning (QML): An Emerging Paradigm

Quantum Machine Learning is an emerging interdisciplinary field that integrates the principles of quantum mechanics with machine learning algorithms. By harnessing the inherent computational power of quantum systems, this field holds the promise of addressing computationally intensive problems that are intractable for classical systems. Quantum Machine Learning is positioned to capitalize on quantum phenomena, such as superposition, entanglement, and interference, to achieve efficiencies that surpass the capabilities of traditional machine learning methods.

Quantum Machine Framework

A quantum computing framework enables data to be encoded and processed according to the principles of quantum mechanics. The computational workflow can be divided into three primary stages:

Data Encoding

Classical data must first be mapped into quantum states, a process essential for the quantum machine to perform computations. This is achieved using techniques such as amplitude encoding, angle encoding, or basis encoding. Amplitude encoding is particularly advantageous due to its ability to compactly represent high-dimensional classical data $\mathbf{x} \in \mathbb{R}^n$ as a quantum state $|\psi\rangle$:

$$|\psi\rangle = \frac{1}{\|\mathbf{x}\|} \sum_{i=1}^{n} x_i |i\rangle \tag{4.5}$$

where $\|\mathbf{x}\|$ normalizes the state, ensuring $|\psi\rangle$ lies on the unit sphere in the Hilbert space.

Quantum Processing

The encoded data is then processed through quantum circuits composed of unitary operations U. These quantum gates operate on qubits, exploiting quantum parallelism and interference to solve problems such as clustering, optimization, or classification. The evolution of the quantum state follows:

$$|\psi_{\text{final}}\rangle = U|\psi_{\text{initial}}\rangle \tag{4.6}$$

Entanglement between qubits enhances computational efficiency, enabling the representation and processing of correlations in high-dimensional data.

Measurement and Decoding

After quantum processing, the final quantum state is measured to extract classical information. The measurement collapses the quantum state $|\psi_{\text{final}}\rangle$ into a classical outcome x, with probabilities determined by the squared amplitudes of the quantum state:

$$P(x) = |\langle x|\psi_{\text{final}}\rangle|^2 \tag{4.7}$$

The measurement process translates quantum computational outcomes into actionable insights for the target ML task.

Advantages of Quantum Machines

Quantum machines exhibit several advantages over classical computational systems:

- **Computational Speed**: Quantum parallelism enables simultaneous operations across all possible states, significantly accelerating algorithms such as matrix inversion and optimization. For example, quantum systems achieve polynomial speedups in solving linear systems of equations via the Harrow-Hassidim-Lloyd (HHL) algorithm.
- **Memory Efficiency**: Compact encoding of high-dimensional data into quantum states mitigates memory bottlenecks encountered in classical systems.
- **Scalability**: Quantum machines are well-suited for processing exponentially large data structures, as n qubits can represent 2^n states.

Challenges in Realizing Quantum Machines

Despite their potential, quantum machines face significant challenges:

- **Noisy Quantum Systems**: Quantum states are susceptible to decoherence, wherein interactions with the environment degrade computational accuracy.
- **Error Correction**: Error correction techniques, such as repetition codes, are critical for mitigating noise and ensuring computational fidelity.
- **Hardware Limitations**: Current quantum hardware is constrained by the number of qubits, coherence time, and gate fidelity.

4.3.3 Quantum Gates

In the proposed algorithm, we utilized a set of quantum gates, including the Hadamard gate, RZ gate, IsingXY gate, CNOT gate, and CSWAP gate. These gates play a crucial role in encoding, feature extraction, and state manipulation. Below, we provide a brief description of each gate and its mathematical intuition.

Hadamard Gate (H)

The Hadamard gate is used to create superposition states, enabling quantum parallelism. It transforms the computational basis states $|0\rangle$ and $|1\rangle$ into an equal superposition:

$$H|0\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle), \quad H|1\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle). \tag{4.8}$$

The matrix representation of the Hadamard gate is:

$$H = \frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}. \tag{4.9}$$

When applied to a general state $|\psi\rangle = a|0\rangle + b|1\rangle$, the Hadamard gate produces:

$$H|\psi\rangle = \frac{1}{\sqrt{2}}\big((a+b)|0\rangle + (a-b)|1\rangle\big). \tag{4.10}$$

RZ Gate (R_Z)

The R_Z gate applies a rotation around the Z-axis of the Bloch sphere, introducing a phase shift. Its matrix representation is:

$$R_Z(\theta) = \begin{pmatrix} e^{-i\theta/2} & 0 \\ 0 & e^{i\theta/2} \end{pmatrix}. \tag{4.11}$$

For a state $|\psi\rangle = a|0\rangle + b|1\rangle$, the action of the R_Z gate is:

$$R_Z(\theta)|\psi\rangle = ae^{-i\theta/2}|0\rangle + be^{i\theta/2}|1\rangle. \tag{4.12}$$

This gate is used in the algorithm to encode angular relationships of features in the data.

IsingXY Gate

The IsingXY gate captures interactions between qubits by applying a unitary transformation that depends on their X- and Y-axis components. Its operation on two qubits is represented as:

$$U_{XY}(\theta) = \exp(-i\theta(X_1X_2 + Y_1Y_2)), \tag{4.13}$$

where X and Y are Pauli matrices. This gate emphasizes correlations between qubits and models higher-order relationships among features.

CNOT Gate (Controlled-NOT)

The CNOT gate is an entangling gate that flips the state of a target qubit $|t\rangle$ if the control qubit $|c\rangle$ is in state $|1\rangle$. The matrix representation of the CNOT gate is:

$$CNOT = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}. \tag{4.14}$$

The action of the gate is given by:

$$CNOT\,|c\rangle\,|t\rangle = |c\rangle\,|t \oplus c\rangle\,, \tag{4.15}$$

where $\oplus$ denotes addition modulo 2. This gate creates entanglement, enabling the representation of higher-order dependencies in the data.

Controlled-SWAP Gate (CSWAP or Fredkin Gate)

The CSWAP gate swaps the states of two target qubits $|\phi\rangle$ and $|\psi\rangle$ conditionally on the state of the control qubit $|c\rangle$. Its operation is defined as:

$$CSWAP\,|c\rangle\,|\phi\rangle\,|\psi\rangle = \begin{cases} |c\rangle\,|\phi\rangle\,|\psi\rangle & \text{if } c = 0, \\ |c\rangle\,|\psi\rangle\,|\phi\rangle & \text{if } c = 1. \end{cases} \tag{4.16}$$

The CSWAP gate is critical for comparing quantum states and is used in the swap test to compute the similarity between quantum-encoded data points.

These quantum operations collectively enable effective encoding, feature extraction, and classification in the quantum domain.

4.3.4 Classical vs Quantum Feature Space

Classical Feature Space

In classical machine learning, data is represented as feature vectors in an n-dimensional Euclidean space, where each point is expressed as:

$$\mathbf{x} = (x_1, x_2, ..., x_n) \in \mathbb{R}^n \tag{4.17}$$

Classical algorithms, such as support vector machines (SVMs) and neural networks, rely on transformations and statistical operations to extract patterns. However, non-linearly separable data often requires kernel methods, which explicitly or approximately map data into higher-dimensional spaces to improve classification accuracy.

Quantum Feature Space

Quantum computing introduces a **Hilbert space representation**, where data is encoded into quantum states using superposition and entanglement. A classical data point x is transformed into a quantum state using a unitary operation:

$$|\psi_x\rangle = U(x)\,|0\rangle \tag{4.18}$$

Unlike classical feature spaces that require complex transformations to capture intricate relationships, quantum feature spaces naturally embed data into exponentially larger spaces, enabling better pattern recognition.Figure 4.2 represents the data seperbality in classical and quantum feature space.

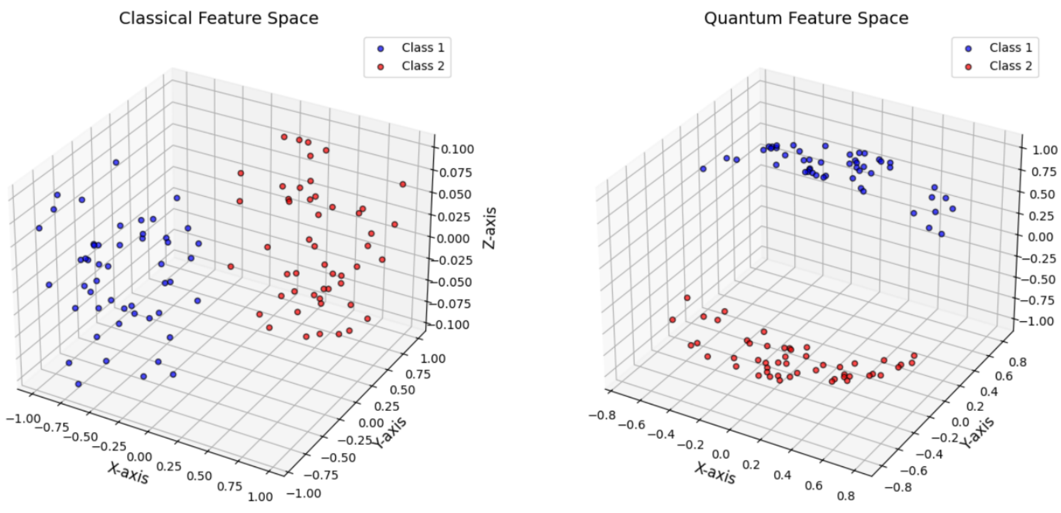


Figure 4.2: Classical Feature Space vs Quantum Feature Space

In contrast to classical feature spaces, which are constrained by computational resources and require extensive feature engineering, quantum feature spaces exploit quantum parallelism to explore a vast state space efficiently, leading to superior classification and clustering performance. This makes quantum machine learning a promising approach for tasks that involve complex high-dimensional data structures.

4.3.5 Quantum State Encoding

Transforming classical data into quantum representations is a crucial step in quantum machine learning approaches, including the proposed Quantum K-Nearest Neighbors algorithm. The aim is to encode classical information within a quantum system, enabling quantum-based techniques to efficiently extract valuable insights. This section investigates the various methods for encoding classical data into quantum states, with an emphasis on the angle encoding technique employed in this investigation. It provides a comparative analysis of alternative encoding approaches, such as amplitude encoding and phase encoding, and discusses their respective strengths, weaknesses, and suitability for different applications within the context of quantum machine learning.

Classical Data Encoding into Quantum States

Quantum encoding transforms classical data points into quantum states, which are vectors in a Hilbert space. Given a classical data vector

$$\mathbf{x} = [x_1, x_2, \ldots, x_d], \tag{4.19}$$

the goal is to map it to a quantum state $|\mathbf{x}\rangle$, where:

$$|\mathbf{x}\rangle = \frac{1}{\|\mathbf{x}\|}\sum_{j=1}^{d} x_j \left|j\right\rangle, \tag{4.20}$$

with $\|\mathbf{x}\|$ being the norm of $\mathbf{x}$. The encoded quantum state must satisfy the normalization condition:

$$\|\mathbf{x}\| = \sqrt{\sum_{j=1}^{d} |x_j|^2} = 1. \tag{4.21}$$

Quantum Data Encoding Techniques

There are different techniques to encode data into quantum states, each with its own advantages and trade-offs. These techniques are:

Amplitude Encoding

Amplitude encoding maps a data vector into the amplitudes of a quantum state, ensuring that the quantum state remains normalized. For a data vector

$$\mathbf{x} = [x_1, x_2, \ldots, x_d] \tag{4.22}$$

the quantum state is encoded as:

$$|\mathbf{x}\rangle = \frac{1}{\|\mathbf{x}\|} \sum_{j=1}^{d} x_j \, |j\rangle \, . \tag{4.23}$$

Advantages:

- Highly efficient in terms of qubit usage, as d-dimensional data requires only $\log_2(d)$ qubits.

Challenges:

- Preparing such states on current quantum hardware is computationally expensive due to the complexity of state preparation circuits.

Basis Encoding

Basis encoding assigns each data point to a basis state of the quantum register. For example, a binary data vector

$$\mathbf{x} = [x_1, x_2, \ldots, x_n] \tag{4.24}$$

maps directly to a computational basis state:

$$|\mathbf{x}\rangle = |x_1 x_2 \ldots x_n\rangle \tag{4.25}$$

Advantages:

- Simple and straightforward to implement.

Challenges:

- Inefficient for continuous-valued or high-dimensional data, as it requires a large number of qubits.

Angle Encoding

Angle encoding represents classical data using the angles of quantum rotation gates. Each data feature x_j is encoded as the rotation angle of a quantum gate, such as $R_y(x_j)$ or $R_z(x_j)$. The state is prepared as:

$$|\mathbf{x}\rangle = \bigotimes_{j=1}^{d} R_y(x_j) |0\rangle , \tag{4.26}$$

where:

$$R_y(x_j) = \exp\left(-i\frac{x_j Y}{2}\right). \tag{4.27}$$

Advantages:

- Efficient for small- to medium-sized datasets.
- Compatible with hardware-efficient circuits on near-term quantum devices.

Challenges:

- Requires additional gates for encoding multi-dimensional data.

Hybrid Encoding

Hybrid encoding combines multiple encoding strategies to optimize performance for specific datasets or hardware constraints. For example, angle encoding can be used for certain features, while amplitude encoding is applied to others.

Advantages:

- Flexibility in handling various types of data distributions.

Challenges:

- Increased circuit complexity and potential overhead in implementation.

The proposed QKNN algorithm utilizes angle encoding to represent the training and test data. This technique involves encoding each feature of the data as the angle of a rotation gate applied to the qubits. This approach is well-suited for current quantum hardware, as it strikes a balance between simplicity and computational efficiency. The encoded quantum states are then processed by the quantum circuit to compute similarity measures, employing techniques such as the swap test.The potential benefits of quantum encoding include improved scaling properties, increased parallelism, and the ability to exploit quantum phenomena like entanglement and superposition to enhance the learning process.

4.3.6 Quantum Distance Measures

The proposed Quantum K-Nearest Neighbors algorithm relies heavily on quantum-based distance measures to assess the similarity between quantum-encoded data points. These quantum distance metrics are computed through quantum operations, enabling efficient and precise evaluation of similarity in high-dimensional data spaces. This section delves into the details of the quantum inner product, the swap test for measuring overlap, and the construction of quantum distance metrics - all of which are fundamental components of the methodological framework underlying the QKNN algorithm. The utilization of quantum distance measures allows the QKNN algorithm to harness the unique properties of quantum systems, such as superposition and entanglement, to achieve enhanced performance compared to classical approaches, particularly in the context of large-scale or high-dimensional datasets.

Swap Test

In quantum computing, the swap test is a powerful quantum subroutine that enables the efficient evaluation of the similarity between two quantum states. This technique involves preparing an auxiliary qubit in the state $|0\rangle$, applying a Hadamard gate, and then performing a controlled-SWAP operation between the auxiliary qubit and the two quantum states being compared. The probability of measuring the auxiliary qubit in the state $|0\rangle$ is directly proportional to the inner product, or similarity, between the two quantum states. The proposed algorithm leverages this swap test to assess the likeness between the test data point's quantum state and the training data points' quantum states. The mathematical formulation of the swap test and the derived quantum distance metric is discussed in the following sections.Below is the detailed step-by-step process involved in the swap test:

Step-1: Initial State Preparation

The initial state of the quantum system consists of two quantum states $|\phi\rangle$ and $|\psi\rangle$, where $|\phi\rangle$ represents the first quantum state and $|\psi\rangle$ represents the second quantum state. These states are placed in a quantum register alongside an auxiliary qubit initialized to $|0\rangle$.

The initial state of the system is:

$$|\psi_{\text{initial}}\rangle = |0\rangle\,|\phi\rangle\,|\psi\rangle \tag{4.28}$$

Step-2: Hadamard Gate on the Auxiliary Qubit

A Hadamard gate is applied to the auxiliary qubit (the first qubit in the state). The Hadamard gate creates a superposition of $|0\rangle$ and $|1\rangle$, which is essential for the quantum interference that will follow.

After applying the Hadamard gate, the state of the system becomes:

$$\frac{1}{\sqrt{2}}\left(|0\rangle\,|\phi,\psi\rangle + |1\rangle\,|\phi,\psi\rangle\right) \tag{4.29}$$

The Hadamard gate has placed the auxiliary qubit into an equal superposition of $|0\rangle$ and $|1\rangle$.

Step-3: Controlled-SWAP Gate

The controlled-SWAP is then applied to the quantum system. This gate performs a swap operation between the states $|\phi\rangle$ and $|\psi\rangle$, depending on the value of the auxiliary qubit. If the auxiliary qubit is in the state $|0\rangle$, the two quantum states are not swapped; if the auxiliary qubit is in state $|1\rangle$, the two quantum states are swapped.

The controlled-SWAP gate transforms the system into the following state:

$$\frac{1}{\sqrt{2}}\left(|0\rangle\,|\phi,\psi\rangle + |1\rangle\,|\psi,\phi\rangle\right) \tag{4.30}$$

Step-4: Second Hadamard Gate on the Auxiliary Qubit

The second Hadamard gate results in:

$$\frac{1}{2}\left(|0\rangle\left(|\phi,\psi\rangle + |\psi,\phi\rangle\right) + |1\rangle\left(|\phi,\psi\rangle - |\psi,\phi\rangle\right)\right) \tag{4.31}$$

Step-5: Measurement of the Auxiliary Qubit

After the second Hadamard Gate Operation, the final step involves measuring the auxiliary qubit (the first qubit). The measurement reveals whether the state of the auxiliary qubit is $|0\rangle$ or $|1\rangle$. The probability of measuring $|0\rangle$ is proportional to the inner product between the two quantum states, $|\phi\rangle$ and $|\psi\rangle$.

The probability of measuring the first qubit in the state $|0\rangle$ is given by:

$$P(\text{First qubit} = 0) = \frac{1}{2}\left(\langle\phi|\langle\psi| + \langle\psi|\langle\phi|\right)\frac{1}{2}\left(|\phi\rangle|\psi\rangle + |\psi\rangle|\phi\rangle\right) = \frac{1}{2} + \frac{1}{2}\left|\langle\psi|\phi\rangle\right|^2 \tag{4.32}$$

If the states $|\phi\rangle$ and $|\psi\rangle$ are orthogonal (i.e., $\langle\phi|\psi\rangle = 0$), the probability of measuring $|0\rangle$ is $\frac{1}{2}$. If the states are identical ($\langle\phi|\psi\rangle = 1$), the probability of measuring $|0\rangle$ is 1.

Step-6:Interpretation of Results

- If the measured probability of finding the auxiliary qubit in state $|0\rangle$ is close to 1, this indicates that the two quantum states are very similar.
- If the measured probability is close to $\frac{1}{2}$, the quantum states are orthogonal.
- The probability gives a direct measure of the similarity (inner product) between the two quantum states.

This process allows us to assess the ”closeness” of quantum states efficiently, making it a powerful tool for quantum state comparison. Using these swap test methodology we measure the similarity between the quantum encoded training and test data points.The proposed distance measure method harnesses the advantages of quantum parallelism to effectively manage large-scale datasets and high-dimensional data, mitigating certain drawbacks inherent in traditional KNN algorithms.

4.3.7 Feature Selection Using Chi-Square Test

The **Chi-Square** (χ^2) test is employed as a feature selection method to determine the most significant categorical variables that influence classification [70]. Feature selection is crucial in QKNN as it reduces computational complexity by eliminating irrelevant or redundant features, thereby improving model efficiency and accuracy. The **Chi-Square feature selection** method evaluates the dependency between each feature and the target class, ranking features based on their contribution to classification. Mathematically, the **Chi-Square statistic** is computed as:

$$\chi^2 = \sum \frac{(O_i - E_i)^2}{E_i} \tag{4.33}$$

where O_i is the observed frequency of a feature in a particular class, and E_i is the expected frequency, given by:

$$E_i = \frac{(\text{Row Total} \times \text{Column Total})}{\text{Grand Total}} \tag{4.34}$$

A higher χ^2 value indicates a stronger dependence between the feature and the target variable, making it more relevant for the QKNN model [71]. Features with low χ^2 values contribute less to classification and are discarded to optimize the quantum state encoding process in QKNN. This selection process ensures that the most informative features are retained, enhancing the **quantum feature encoding** and reducing noise, ultimately improving the classification performance of QKNN.

4.4 Proposed Algorithm Design:Quantum K-Nearest Neighbors (QKNN)

In this section, we outline the design of a novel quantum-inspired K-Nearest Neighbors algorithm. This approach leverages quantum computing principles to enhance the performance of the classical KNN algorithm. The QKNN algorithm incorporates quantum circuits and measurement techniques to accelerate the computation of distances, neighbor identification, and the overall classification process. The proposed method utilizes a quantum device to classify test data points based on the nearest neighbors found within the quantum-enriched feature space. By harnessing the unique properties of quantum systems, such as superposition and entanglement, the QKNN algorithm can achieve improved accuracy and efficiency compared to traditional KNN methods, making it a promising approach for a variety of machine learning and data analysis applications.Figure 4.3 represents the methodology of our proposed QKNN algorithm.

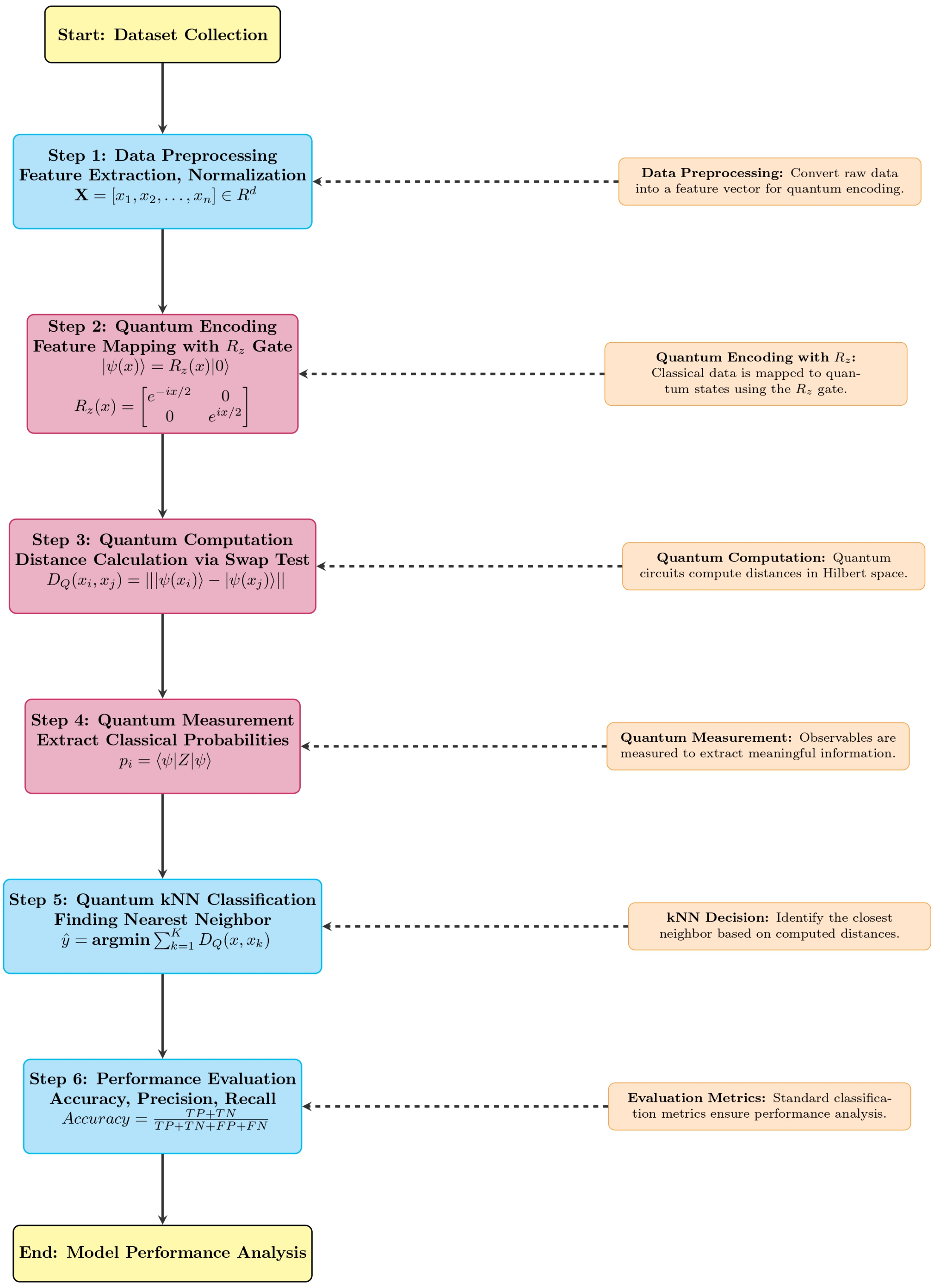

Start: Dataset Collection
Step 1: Data Preprocessing
Feature Extraction, Normalization
$\mathbf{X} = [x_1, x_2, \ldots, x_n] \in R^d$
Data Preprocessing: Convert raw data into a feature vector for quantum encoding.
Step 2: Quantum Encoding
Feature Mapping with R_z Gate
$|\psi(x)\rangle = R_z(x)|0\rangle$
$R_z(x) = \begin{bmatrix} e^{-ix/2} & 0 \\ 0 & e^{ix/2} \end{bmatrix}$
Quantum Encoding with R_z: Classical data is mapped to quantum states using the R_z gate.
Step 3: Quantum Computation
Distance Calculation via Swap Test
$D_Q(x_i, x_j) = |||\psi(x_i)\rangle - |\psi(x_j)\rangle||$
Quantum Computation: Quantum circuits compute distances in Hilbert space.
Step 4: Quantum Measurement
Extract Classical Probabilities
$p_i = \langle\psi|Z|\psi\rangle$
Quantum Measurement: Observables are measured to extract meaningful information.
Step 5: Quantum kNN Classification
Finding Nearest Neighbor
$\hat{y} = \text{argmin} \sum_{k=1}^{K} D_Q(x, x_k)$
kNN Decision: Identify the closest neighbor based on computed distances.
Step 6: Performance Evaluation
Accuracy, Precision, Recall
$Accuracy = \frac{TP+TN}{TP+TN+FP+FN}$
Evaluation Metrics: Standard classification metrics ensure performance analysis.
End: Model Performance Analysis

Figure 4.3: Q-KNN Methodology

4.4.1 Input Representation: Encoding Classical Data into Quantum States

The initial step in the QKNN algorithm involves encoding classical data into quantum states, as quantum computers inherently operate on quantum information. The training dataset $X_{\text{train}} =$

$\{x_{\text{train}}^{(1)}, x_{\text{train}}^{(2)}, \dots, x_{\text{train}}^{(N)}\}$ and the test dataset $X_{\text{test}} = \{x_{\text{test}}^{(1)}, x_{\text{test}}^{(2)}, \dots\}$ are systematically transformed into quantum states through the application of quantum operations such as Hadamard gates and R_Z rotation gates.

The encoding process begins with the application of Hadamard gates H to the quantum register to prepare it in a superposition state. A Hadamard gate applied to a single qubit transforms the computational basis state $|0\rangle$ into an equal superposition of $|0\rangle$ and $|1\rangle$:

$$H|0\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle) \tag{4.35}$$

When applied to all d qubits in the register, where d represents the number of features in the data point, the Hadamard operation creates a superposition of all possible basis states:

$$|\psi_{\text{init}}\rangle = \frac{1}{\sqrt{2^d}} \sum_{i=0}^{2^d-1} |i\rangle \tag{4.36}$$

This superposition state enables the quantum computer to leverage quantum parallelism in subsequent computations, which is critical for efficient distance evaluations across multiple training data points.

After initializing the superposition state, the classical data features are encoded into the quantum register through the application of R_Z rotation gates. For a given feature x_i, the rotation gate $R_Z(\theta)$ is defined as:

$$R_Z(\theta) = \begin{bmatrix} e^{-i\theta/2} & 0 \\ 0 & e^{i\theta/2} \end{bmatrix} \tag{4.37}$$

where θ is proportional to the normalized feature value x_i. A common normalization approach is scaling $\theta = 2\pi x_i$ for feature values within $[0, 1]$. The application of $R_Z(\theta)$ to a qubit transforms it into the state:

$$|\psi_i\rangle = R_Z(\theta_i)H|0\rangle = \frac{1}{\sqrt{2}}\left(e^{-i\theta_i/2}|0\rangle + e^{i\theta_i/2}|1\rangle\right) \tag{4.38}$$

For a train data point with d features, the complete quantum state encoding is given by:

$$|x_{\text{train}}\rangle = |\psi_1\rangle \otimes |\psi_2\rangle \otimes \cdots \otimes |\psi_d\rangle \tag{4.39}$$

This process effectively maps each classical data point into a quantum state, preserving the structure and distribution of the input data in the quantum Hilbert space. For the test data points similar procedure is used.

4.4.2 Feature Extraction using Quantum Gates

After the classical data is encoded into quantum states, quantum gates are applied to extract relevant features from the data. Quantum gates manipulate quantum states through unitary transformations, which are mathematically described by unitary matrices. These operations preserve the normalization of quantum states while emphasizing or extracting specific features from the encoded data. In the context of quantum machine learning, these transformations can achieve several important objectives, such as introducing non-linear mappings, embedding data into higher-dimensional spaces, and reducing data overlap for improved classification.

Non-linear Transformations

Quantum gates, like rotation gates, introduce non-linear transformations that can capture complex relationships between features. For example, the rotation gate $R_Z(\theta)$ acts on a qubit by rotating its state vector around the z-axis of the Bloch sphere by an angle θ. This transformation is expressed mathematically as:

$$R_Z(\theta) = \begin{bmatrix} e^{-i\theta/2} & 0 \\ 0 & e^{i\theta/2} \end{bmatrix} \tag{4.40}$$

When applied to a quantum state $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$, the resulting state becomes:

$$|\psi_{\text{new}}\rangle = R_Z(\theta)|\psi\rangle = \alpha e^{-i\theta/2}|0\rangle + \beta e^{i\theta/2}|1\rangle \tag{4.41}$$

This rotation introduces a phase shift in the amplitudes of the quantum state, emphasizing angular relationships in the data. When such gates are applied to multiple qubits, they create non-linear transformations that can reveal complex patterns and correlations in the data, patterns that are challenging for classical algorithms to identify. Figure 4.4 represents the circuit architecture which is used in our proposed algorithm.

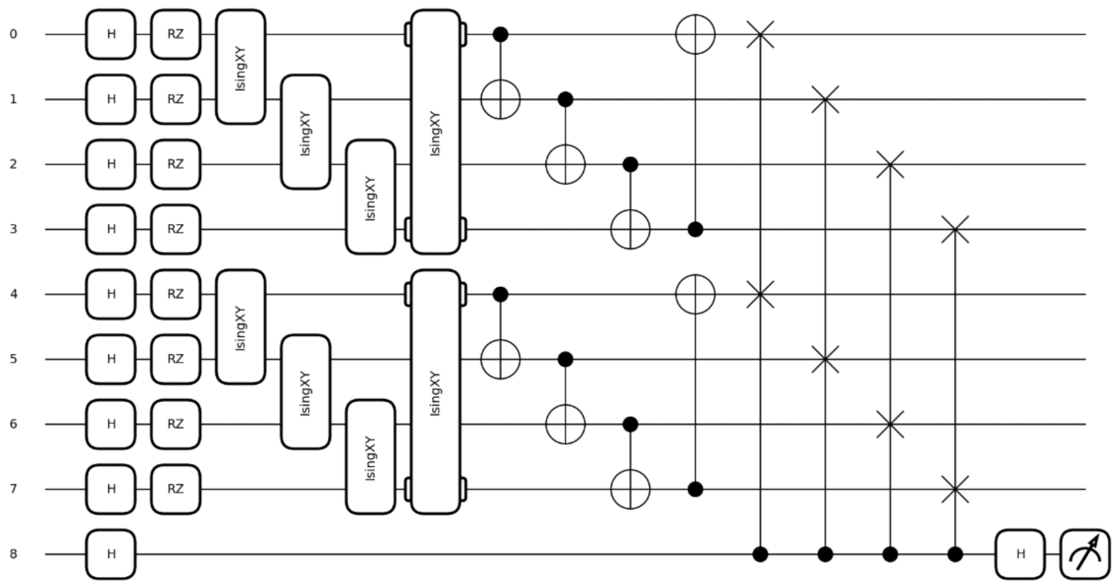


Figure 4.4: Q-KNN architecture

Higher-dimensional Embedding

Quantum systems operate in an exponentially larger Hilbert space compared to classical systems. Specifically, for a system of n qubits, the quantum state resides in a Hilbert space of dimension 2^n. This higher-dimensional representation allows quantum computers to encode more information and capture relationships in the data that may be hidden in lower-dimensional spaces.

For example, applying an IsingXY gate to a pair of qubits creates entanglement and enhances their correlations. The IsingXY gate is represented as:

$$U_{\text{IsingXY}}(\theta) = \exp\left(-i\frac{\theta}{2}(\sigma_x^{(1)}\sigma_x^{(2)} + \sigma_y^{(1)}\sigma_y^{(2)})\right) \tag{4.42}$$

where σ_x and σ_y are the Pauli operators. This entangling operation modifies the quantum state of the qubits, capturing higher-order interactions between features in a way that is not feasible classically. The entanglement introduced by such gates can lead to better separability of classes in the feature space, improving the performance of classification tasks.

Reduction of Data Overlap

Another key benefit of quantum gates is their ability to reduce the overlap between data points, which enhances the distinctness of different classes. Gates like the CNOT (Controlled-NOT) gate play a critical role in this aspect. When applied to two qubits, the CNOT gate flips the second qubit (target qubit) conditionally based on the state of the first qubit (control qubit). Mathematically, the action of the CNOT gate on a two-qubit state $|q_1\rangle = \alpha|0\rangle + \beta|1\rangle$ and $|q_2\rangle = \gamma|0\rangle + \delta|1\rangle$ is:

$$|q_1, q_2\rangle \to \text{CNOT}|q_1, q_2\rangle = \alpha|00\rangle + \beta|01\rangle + \gamma|11\rangle + \delta|10\rangle \tag{4.43}$$

By entangling the two qubits, the CNOT gate increases the "distance" between quantum states that are close together, making it easier to distinguish between data points. This reduction of overlap in the quantum feature space helps to improve classification accuracy by ensuring that different classes are more distinct and easier to separate.

In the proposed algorithm, a combination of R_Z gates, IsingXY gates, and CNOT gates was applied to both the training and test datasets. These quantum gates were used to transform the quantum-encoded data, ensuring that both datasets underwent the same feature extraction process. This transformation can be summarized mathematically as:

$$|\psi_{\text{train}}\rangle \to U|\psi_{\text{train}}\rangle \tag{4.44}$$

$$|\psi_{\text{test}}\rangle \to U|\psi_{\text{test}}\rangle \tag{4.45}$$

where U represents the composite unitary operation resulting from the sequence of quantum gates applied to the data. By applying the same set of quantum operations to both the training and test data, the algorithm ensures that the data is consistently transformed, allowing for meaningful comparisons between the datasets in the quantum feature space. This consistency is critical for effective classification, as it ensures that the features of both datasets are mapped to the same quantum space, enabling accurate distance calculations and class predictions.

4.4.3 Distance Calculation and Neighbor Selection

In the proposed Quantum K-Nearest Neighbors (QKNN) algorithm, the core task of distance calculation is executed through quantum operations within a quantum circuit. This step quantifies the similarity between a test data point and each point in the training set. The quantum circuit, through its execution, generates quantum states that represent the data, and the distance between these quantum states serves as the primary measure of similarity.

Quantum Distance Metric and the Swap Test

The distance between quantum states is inferred from the output of the quantum circuit, which computes the expectation value of the Pauli-Z operator. This expectation value provides insight into the similarity between two quantum-encoded data points. The Pauli-Z operator Z applied to a quantum state $|\psi\rangle$ yields a value between 0 and 1, where a value close to 1 indicates that the states are highly similar, and a value closer to 0.5 suggests that they are dissimilar.

To calculate the quantum distance between two quantum states $|\psi_{\text{train},i}\rangle$ (representing a training data point) and $|\psi_{\text{test}}\rangle$ (representing a test data point), we use the *swap test*. The swap test is a quantum protocol that estimates the inner product between two quantum states $|\psi_{\text{train},i}\rangle$ and $|\psi_{\text{test}}\rangle$, which can be related to the quantum distance metric.

The quantum distance D between two quantum states $|\psi_{\text{train},i}\rangle$ and $|\psi_{\text{test}}\rangle$ is formally defined using the swap test function:

$$D(\psi_{\text{train},i}, \psi_{\text{test}}) = \frac{1}{2}\left(1 + |\langle\psi_{\text{test}}|\psi_{\text{train},i}\rangle|^2\right) \tag{4.46}$$

where $|\langle\psi_{\text{test}}|\psi_{\text{train},i}\rangle|^2$ represents the overlap between the two quantum states. The swap test provides an estimate of this overlap, which can be interpreted as a measure of similarity between the quantum states. The swap test involves applying a controlled-SWAP gate and measuring an ancillary qubit, which yields a probability related to the overlap of the quantum states. A higher overlap leads to a higher probability of measuring the ancillary qubit in the $|0\rangle$ state, indicating greater similarity between the states.

In practice, the *swap test* allows the estimation of the overlap between the quantum states of the training and test data points. The quantum distance between the states is computed based on this overlap, with smaller distances indicating higher similarity between the points.

Neighbor Selection

Once the quantum distances are computed for each pair of test and training data points, the next step in the QKNN algorithm is the *selection of the nearest neighbors*. This step identifies the K most similar training points to a given test data point. The selection of neighbors is based on the quantum distance metric, where smaller distances indicate higher similarity.

For each test point $X_{\text{test},j}$, the quantum distances to all training points $X_{\text{train},i}$ are calculated using the quantum distance formula derived from the swap test:

$$D(X_{\text{test},j}, X_{\text{train},i}) = \frac{1}{2}\left(1 + |\langle\psi_{\text{test}}|\psi_{\text{train},i}\rangle|^2\right) \tag{4.47}$$

The K-nearest neighbors are selected by choosing the K training points with the smallest quantum distances, i.e., the K training points that minimize the quantum distance metric.

The *class prediction* for the test point $X_{\text{test},j}$ is determined using majority voting over the class labels of the selected nearest neighbors. Let the class labels of the K-nearest neighbors be denoted as $C_{\text{train},i}$. The class label C_{pred} assigned to the test point is given by:

$$C_{\text{pred}}(X_{\text{test},j}) = \arg\max_{C_i} \sum_{i=1}^{K} \mathbb{I}(C_{\text{train},i} = C_{\text{train},i}) \tag{4.48}$$

where $\mathbb{I}(\cdot)$ is the indicator function that returns 1 if the condition inside the parentheses is true, and 0 otherwise. In other words, the test point is classified into the class that appears most frequently among the K-nearest neighbors.

Algorithm 6 Quantum K-Nearest Neighbors (QKNN) Algorithm

```
Require: Training dataset $X_{\text{train}}, Y_{\text{train}}$, test instance $X_{\text{test}}$, number of neighbors $K$
Ensure: Predicted class label $C_{\text{pred}}$
1: Encode classical data into quantum states using Hadamard and rotation gates.
2: Apply quantum transformations (IsingXV, CNOT) for feature extraction.
3: Compute quantum distances using the swap test.
4: Select $K$ nearest neighbors based on minimum quantum distances.
5: Determine the class using majority voting among selected neighbors.
6: return $C_{\text{pred}} = 0$
```

4.4.4 Advantages and Limitations of the Proposed Quantum k-NN Algorithm

Advantages

1. Quantum Parallelism for Faster Distance Computation

The variational quantum circuit enables simultaneous distance calculations in Hilbert space, reducing computational complexity compared to classical k-NN. Unlike classical methods, which require $O(n)$ computations for distance measurement, quantum circuits can potentially reduce this to $O(\log n)$ under ideal conditions.

2. Feature-Rich Quantum Encoding Using R_z Gates

The use of R_z gates for encoding allows the model to capture phase-based feature interactions, making the quantum feature space more expressive. The rotation encoding matrix is given by:

$$R_z(x) = \begin{bmatrix} e^{-ix/2} & 0 \\ 0 & e^{ix/2} \end{bmatrix} \tag{4.49}$$

This transformation enhances the model's ability to separate data points in higher-dimensional quantum space.

3. Improved Classification Accuracy in High-Dimensional Spaces
By leveraging the quantum Hilbert space, the quantum distance metric (e.g., Fubini-Study or Hilbert-Schmidt) provides superior class separation compared to Euclidean distance in classical k-NN.

4. Potential Quantum Speedup in Large Datasets
Quantum computation offers logarithmic improvements in computational complexity for large-scale classification problems, making it highly scalable for high-dimensional medical datasets.

5. Robust to Small Datasets
Unlike deep learning models, which require extensive training data, QkNN can work effectively with limited datasets, such as medical diagnosis problems where labeled data is scarce.

Limitations

1. Hardware Limitations: Noise and Decoherence
Current Noisy Intermediate-Scale Quantum (NISQ) devices introduce quantum noise, which can reduce classification accuracy. Decoherence and gate errors can affect the stability of quantum states.

2. Encoding Overhead and Qubit Requirements
Encoding high-dimensional classical features into quantum states requires a substantial number of qubits and repeated quantum state preparations, increasing the circuit complexity.

3. Measurement Overhead and Readout Errors
Quantum measurements collapse quantum states, necessitating multiple repeated executions for stable classification results. Readout errors can further impact reliability.

4. Lack of a Proven Quantum Speedup in Practical Scenarios
While theoretical speedups exist, real-world advantages depend on specific problem structures and noise-resilient quantum hardware.

5. High Computational Cost of Quantum State Preparation
Encoding complex datasets may require a significant number of quantum gates, offsetting the benefits of quantum parallelism.

6. Dependency on Hybrid Quantum-Classical Optimization
Hybrid models require both quantum circuits and classical optimization, which can introduce additional computational overhead and slow convergence.

4.5 Simulation or Implementation Details

The simulation of quantum algorithms in this work utilized the Pennylane framework, a versatile and widely adopted platform for hybrid quantum-classical computations. Pennylane facilitated seamless integration between quantum and classical computations, enabling efficient implementation and testing of quantum algorithms, particularly in scenarios involving parameterized quantum circuits and variational algorithms.

The lightning.qubit device was employed as the backend simulator. This state-of-the-art, high-performance simulator is based on state-vector methods and written in optimized C++. It is specifically designed for efficient numerical simulation of quantum circuits, providing significant speedup and precision compared to general-purpose simulators. The lightning.qubit device supports advanced features such as customizable circuit operations, precision control, and compatibility with various quantum gradient computation techniques, which are integral to the development of variational algorithms.

The use of the lightning.qubit simulator allowed for comprehensive testing under idealized conditions, facilitating precise exploration of quantum state dynamics and algorithmic behavior. This setup was chosen not only for its computational efficiency but also for its ability to simulate systems with a larger number of qubits, thus enabling scalability analyses critical to the proposed research. Additionally, it offered insights into the potential real-world performance of the algorithms, forming a foundational step before deployment on physical quantum hardware.

This simulation environment was instrumental in validating the theoretical constructs of the proposed quantum algorithms, performing detailed error analyses, and iteratively refining the design based on empirical observations. Using the strengths of Pennylane and the lightning.qubit device, the research benefited from a robust and reliable framework for the development of quantum algorithms. Figures 4.5 and 4.6 show the simulation softwares used in my thesis.

Figure 4.5: Qiskit - IBM's Open-Source Quantum Computing Framework. It is used for simulating quantum circuits and executing them on IBM Quantum hardware.

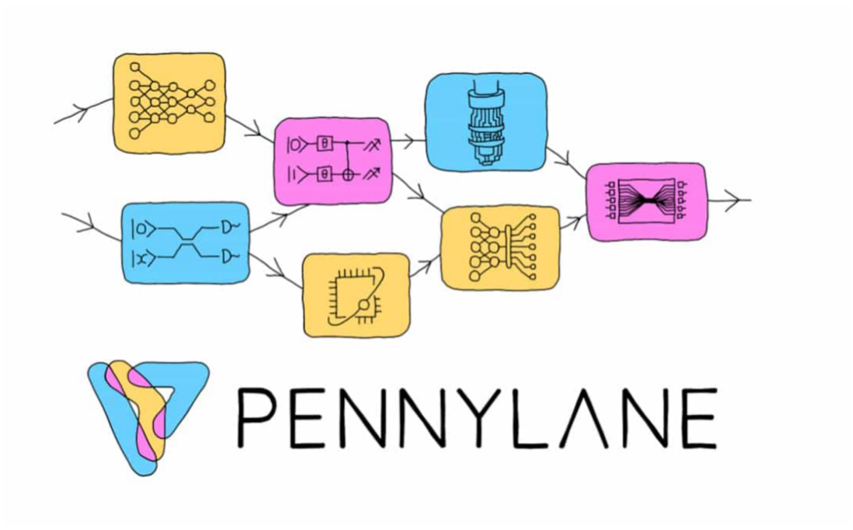


Figure 4.6: PennyLane - A Quantum Machine Learning and Simulation Platform. It is used for quantum computing simulations, variational quantum algorithms, and integrating quantum computations with machine learning.

4.6 Noise Model and Simulation in Q-KNN

Quantum systems are inherently susceptible to noise due to decoherence and imperfections in quantum hardware. One of the fundamental challenges in quantum computing is mitigating these noise effects to ensure reliable computation. In this study, we analyze the impact of different quantum noise channels on the performance of a Quantum K-Nearest Neighbors (QKNN) classifier. The noise model in our quantum circuit incorporates bit-flip, phase-flip, and combined bit-phase flip errors, which are modeled using the Pauli error channels.

4.6.1 Quantum Noise Channels

Quantum noise channels describe the probabilistic evolution of quantum states under noise. The noise applied in our simulation consists of three primary channels:

Bit-Flip Channel (Pauli-X Error)

The bit-flip error is modeled by the application of the Pauli-X gate, which flips the computational basis states:

$$X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}. \tag{4.50}$$

This error transforms to and vice versa:

$$X|0\rangle = |1\rangle, \quad X|1\rangle = |0\rangle. \tag{4.51}$$

The corresponding quantum channel representation is given by:

$$\mathcal{E}_X(\rho) = (1-p)\rho + pX\rho X^\dagger, \tag{4.52}$$

where is the probability of the bit-flip occurring, and is the quantum state.

Phase-Flip Channel (Pauli-Z Error)

The phase-flip error is described by the application of the Pauli-Z gate, which introduces a relative phase shift in the superposition states:

$$Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}. \tag{4.53}$$

The action of the phase-flip error on basis states is:

$$Z|0\rangle = |0\rangle, \quad Z|1\rangle = -|1\rangle. \tag{4.54}$$

The corresponding quantum channel representation is:

$$\mathcal{E}_Z(\rho) = (1-p)\rho + pZ\rho Z^\dagger. \tag{4.55}$$

Bit-Phase Flip Channel (Pauli-Y Error)

A simultaneous bit-flip and phase-flip error corresponds to the Pauli-Y gate:

$$Y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}. \tag{4.56}$$

Applying the Pauli-Y gate to a qubit induces both an inversion and a phase change:

$$Y|0\rangle = i|1\rangle, \quad Y|1\rangle = -i|0\rangle. \tag{4.57}$$

The corresponding quantum channel representation is:

$$\mathcal{E}_Y(\rho) = (1-p)\rho + pY\rho Y^\dagger. \tag{4.58}$$

4.6.2 Implementation of Noise in the Quantum Circuit

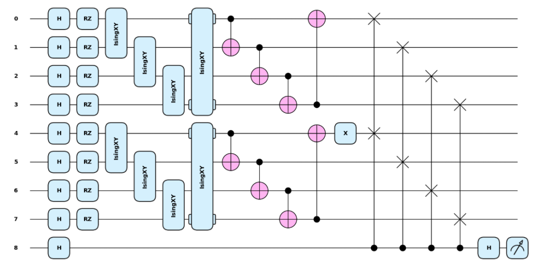
In our simulation, we introduce these noise models probabilistically to each wire in the quantum circuit. For a given error probability , the noise application follows these probabilistic rules:

- With probability $\frac{p}{3}$, a **bit-flip error** (Pauli-X) is applied.
- With probability $\frac{p}{3}$, a **phase-flip error** (Pauli-Z) is applied.
- With probability $\frac{p}{3}$, a **combined bit-flip and phase-flip error** (Pauli-Y) is applied.

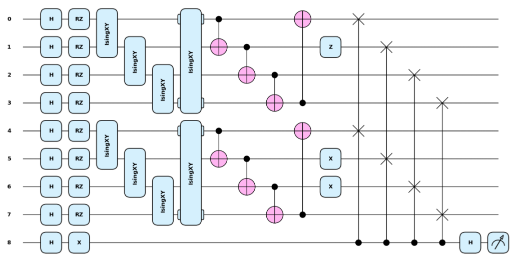
Mathematically, the evolution of the quantum state under this noise model can be expressed as:

$$\mathcal{E}(\rho) = (1-p)\rho + \frac{p}{3}X\rho X + \frac{p}{3}Z\rho Z + \frac{p}{3}Y\rho Y. \tag{4.59}$$

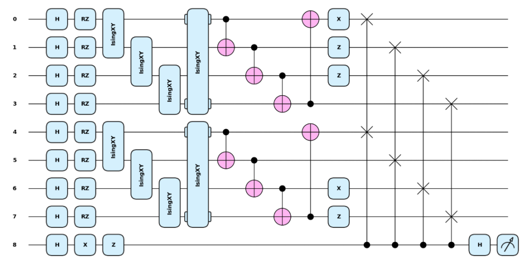
This noise model is implemented in our quantum circuit by iterating over all qubits and stochastically applying these errors based on their assigned probabilities.Figure 4.7 represents how the QKNN architecture circuit affected at different noise probability.



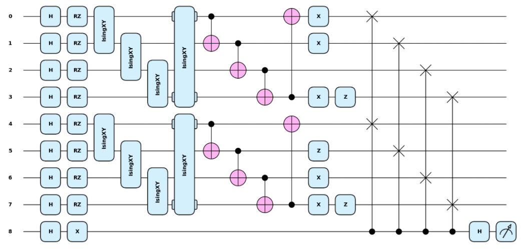
(a) Noise probability $p = 0.1$



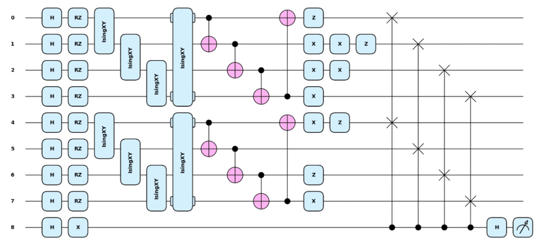
(b) Noise probability $p = 0.2$



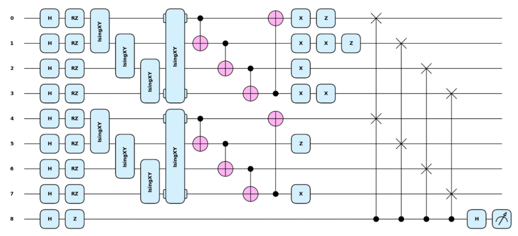
(c) Noise probability $p = 0.3$



(d) Noise probability $p = 0.4$



(e) Noise probability p= 0.5



(f) Noise probability p= 0.6

Figure 4.7: Quantum circuits representing different noise models with increasing Noise probability.

Since QKNN relies on quantum state preparation and distance computation in the Hilbert space, noise introduced at various stages of the circuit can drastically impact classification performance. In particular, errors in the measurement qubit can lead to severe misclassifications, as this qubit determines the final readout of the classifier. The analysis of accuracy degradation as a function of noise probability provides insights into the robustness of QKNN and the necessity of quantum error mitigation techniques for real-world deployment.

4.7 Introduction of Repetition Code for Error Mitigation

In the context of quantum computing, errors frequently arise due to the inherent noise present in quantum devices, which can significantly impact the performance of algorithms. Particularly, bit-flip and phase-flip errors are common occurrences, stemming from imperfect gate operations and decoherence. To mitigate these errors, repetition codes were selected in this work due to their simplicity, efficiency, and effectiveness in correcting such errors. The redundancy-based structure of these codes enables the detection and correction of single-qubit errors without requiring complex quantum error correction protocols. Furthermore, the seamless integration of repetition codes with the Quantum K-Nearest Neighbors algorithm provides a robust solution for maintaining the reliability of quantum computations, which is crucial in noisy intermediate-scale quantum devices. The general applicability and minimal overhead of repetition codes make them an ideal choice for error correction in QKNN, a quantum machine learning algorithm that can benefit from error-resilient data encoding and state processing.

Generalization to n-Qubit Repetition Code

The n-qubit repetition code is a widely employed error correction approach in the realm of quantum computing, distinguished by its simplicity and efficency in mitigating single-qubit errors. This technique involves the redundant encoding of logical qubits across n physical qubits (where n is an odd integer), thereby ensuring the robustness of the logical state against errors induced by environmental noise.

1. Encoding

The core idea of the n-qubit repetition code is to redundantly encode a single logical qubit across n physical qubits to preserve the logical information. The encoding process is defined as follows:

- A logical $|0\rangle$ is encoded as:

$$|0\rangle_L = |0\rangle^{\otimes n} = |00\ldots 0\rangle \tag{4.60}$$

  where $|0\rangle^{\otimes n}$ denotes n qubits all in the state $|0\rangle$.

- A logical $|1\rangle$ is encoded as:

$$|1\rangle_L = |1\rangle^{\otimes n} = |11\ldots 1\rangle \tag{4.61}$$

- For a general logical qubit $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$, the encoding is given by:

$$|\psi\rangle_L = \alpha|0\rangle^{\otimes n} + \beta|1\rangle^{\otimes n} \tag{4.62}$$

This redundant encoding ensures that the logical state $|\psi\rangle$ is distributed across multiple physical qubits, making it resilient to errors on any single qubit.

2. Error Detection

Errors are detected by measuring stabilizers, which assess the parity between adjacent qubits. Stabilizers for the n-qubit code are defined as:

$$S_i = Z_i Z_{i+1}, \quad i \in \{1, 2, \ldots, n-1\} \tag{4.63}$$

where Z_i is the Pauli-Z operator acting on the i-th qubit. These stabilizers compare the states of neighboring qubits and produce measurement outcomes of either $+1$ or -1:

- **Stabilizer outcome** $+1$: The two adjacent qubits have the same state (e.g., both $|0\rangle$ or both $|1\rangle$).
- **Stabilizer outcome** -1: The two adjacent qubits are in different states (e.g., one is $|0\rangle$ and the other is $|1\rangle$).

By analyzing the sequence of stabilizer outcomes (the syndrome), errors can be localized to specific qubits.

3. Error Correction

Once an error is detected, the logical state is corrected based on the stabilizer syndromes. The error correction procedure involves the following steps:

1. **Identifying the erroneous qubit:** The stabilizer outcomes indicate which adjacent qubits have mismatched states, pinpointing the location of the error.
2. **Correcting the error:** A Pauli-X gate (X_i) is applied to the erroneous qubit, flipping its state to restore parity with its neighbors.The error syndrome table 4.1 is used to correct errors.

Table 4.1: Error Syndrome Table for 3-Qubit Repetition Code

S_1	S_2	**Error Pattern**	**Correction**
$+1$	$+1$	$I \otimes I \otimes I$	$No\, Error$
$+1$	-1	$I \otimes I \otimes X$	$Apply\, X\, gate\, on\, Qubit1$
-1	$+1$	$X \otimes I \otimes I$	$Apply\, X\, gate\, on\, Qubit3$
-1	-1	$I \otimes X \otimes I$	$Apply\, X\, gate\, on\, Qubit2$

3. **Majority voting:** To determine the final logical state, the majority voting rule is applied:

- If the majority of qubits are in state $|0\rangle$, the logical state is $|0\rangle_L$.
- If the majority of qubits are in state $|1\rangle$, the logical state is $|1\rangle_L$.

This approach ensures that single-qubit errors are corrected, preserving the integrity of the logical state.

Chapter 5

Results and Discussion

This section presents the outcomes of applying both Quantum-KNN (Q-KNN) and Classical-KNN (C-KNN) across three datasets—Breast Cancer, Iris, and Bank Note—and offers a detailed discussion. Key performance metrics are highlighted, and confusion matrices are noted to further elucidate each model's classification behavior.

Performance metrics used to evaluate our proposed model are:

Confusion Matrix. In a classification setting, outcomes can be summarized in a confusion matrix as shown in Table 5.1:

Table 5.1: Basic confusion matrix for classification.

	Predicted	
	Positive	**Negative**
Actual Positive	True Positive (TP)	False Negative (FN)
Actual Negative	False Positive (FP)	True Negative (TN)

Each metric below highlights different aspects of classification performance:

- **Accuracy:**
$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \tag{5.1}$$
Reflects the proportion of correctly classified instances among all predictions.

- **Precision:**
$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{5.2}$$
Indicates how often predicted positives are correct.

- **Recall (Sensitivity):**
$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{5.3}$$

Measures how effectively the model identifies actual positives.

- **F1-Score:**
$$F1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{5.4}$$
Balances precision and recall into a single measure.

- **Area Under the Curve (AUC):** Refers to the area under the Receiver Operating Characteristic (ROC) curve, which plots the True Positive Rate (Recall) against the False Positive Rate. Higher AUC values indicate better overall discrimination across different thresholds.

Table 5.2: Comparison of Proposed Q-KNN and KNN Accuracies Across Datasets

Dataset	**Proposed Q-KNN Accuracy**	**KNN Accuracy**
Breast Cancer	0.9825	0.9298
Iris	1.0000	1.0000
Bank Note	0.9927	0.9855

As shown in Table 5.2, Q-KNN consistently outperforms or matches the accuracy of classical KNN. For the breast cancer data set, Q-KNN achieves an accuracy of 0.9825, significantly exceeding KNN (0.9298), indicating an enhanced classification capacity. In the Iris dataset, both models achieve a perfect accuracy of 1.0000, suggesting that the data set is linearly separable. For the Bank Note dataset, Q-KNN attains 0.9927, slightly outperforming KNN (0.9855), demonstrating improved discrimination ability.

Table 5.3: Performance Metrics of Proposed Quantum-KNN (Q-KNN) Architecture

Dataset	**AUC**	**F1-Score**	**Precision**	**Recall**
Breast Cancer	0.9859	0.98	0.98	0.99
Iris	1.0000	1.00	1.00	1.00
Bank Note	0.9921	0.99	0.99	0.99

Table 5.3 further validates the effectiveness of Q-KNN. The model achieves high AUC values (0.9859, 1.0000, and 0.9921 for breast cancer, iris, and banknote, respectively), indicating strong classification ability. Additionally, the F1-scores (0.98, 1.00, and 0.99) confirm that Q-KNN maintains a balance between precision and recall, minimizing misclassification. High precision and recall values across all datasets further reinforce the reliability of Q-KNN in classifying instances with minimal errors.Figure 5.4 shows the confusion matrix for different datasets and Figure 5.5 shows the performance comparison of Q-KNN and KNN model.

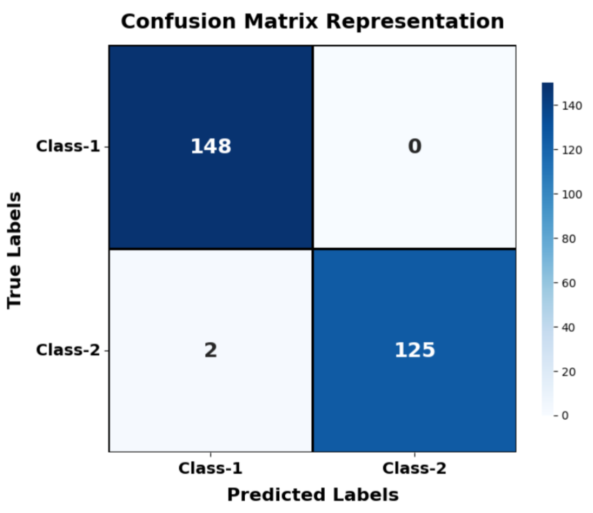


Figure 5.1: Confusion Matrix for Banknote Authentication dataset

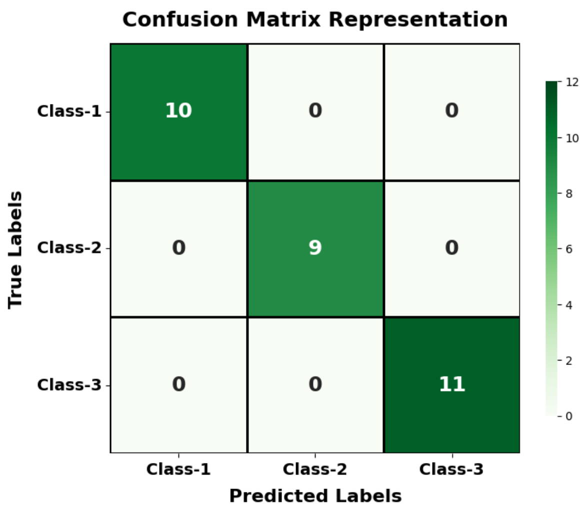


Figure 5.2: Confusion Matrix for Iris dataset

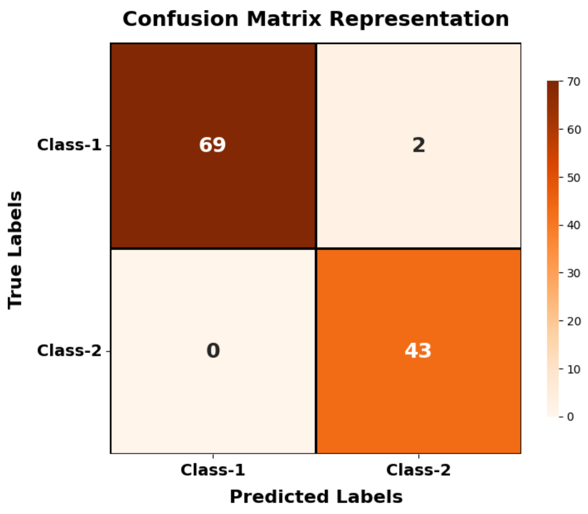


Figure 5.3: Confusion Matrix for Breast Cancer Wisconsin dataset

Figure 5.4: Confusion Matrices for Different Datasets

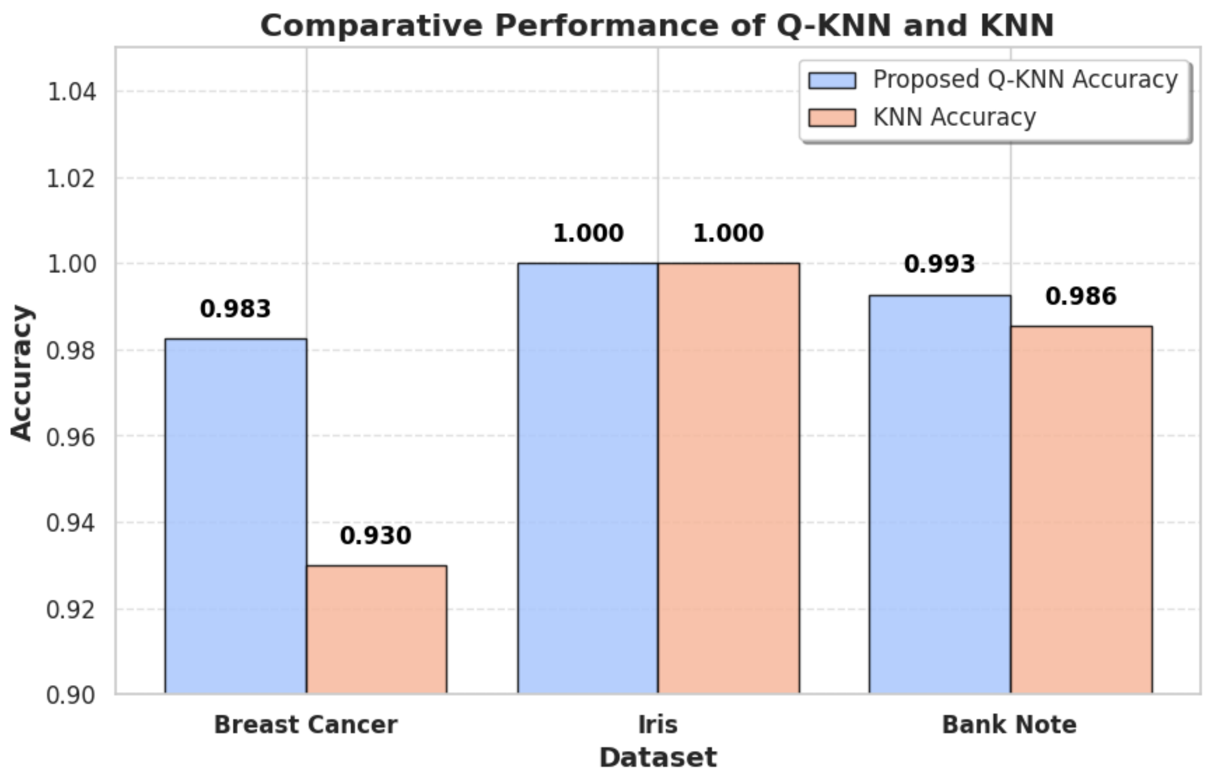


Figure 5.5: Comparitive Performance of proposed Q-KNN and KNN

The superior performance of Q-KNN can be attributed to the advantages of quantum computing, such as superposition and entanglement, which enhance pattern recognition. The model demonstrates significant improvements, particularly for complex datasets such as Breast Cancer and Bank Note, where classical KNN exhibits limitations. These results highlight Q-KNN as a promising alternative for machine learning applications, particularly in biomedical and financial domains where high accuracy is critical. Future research could explore its scalability and applicability to more complex datasets.Figure 5.6 shows the impact of noise probability in model accuracy.

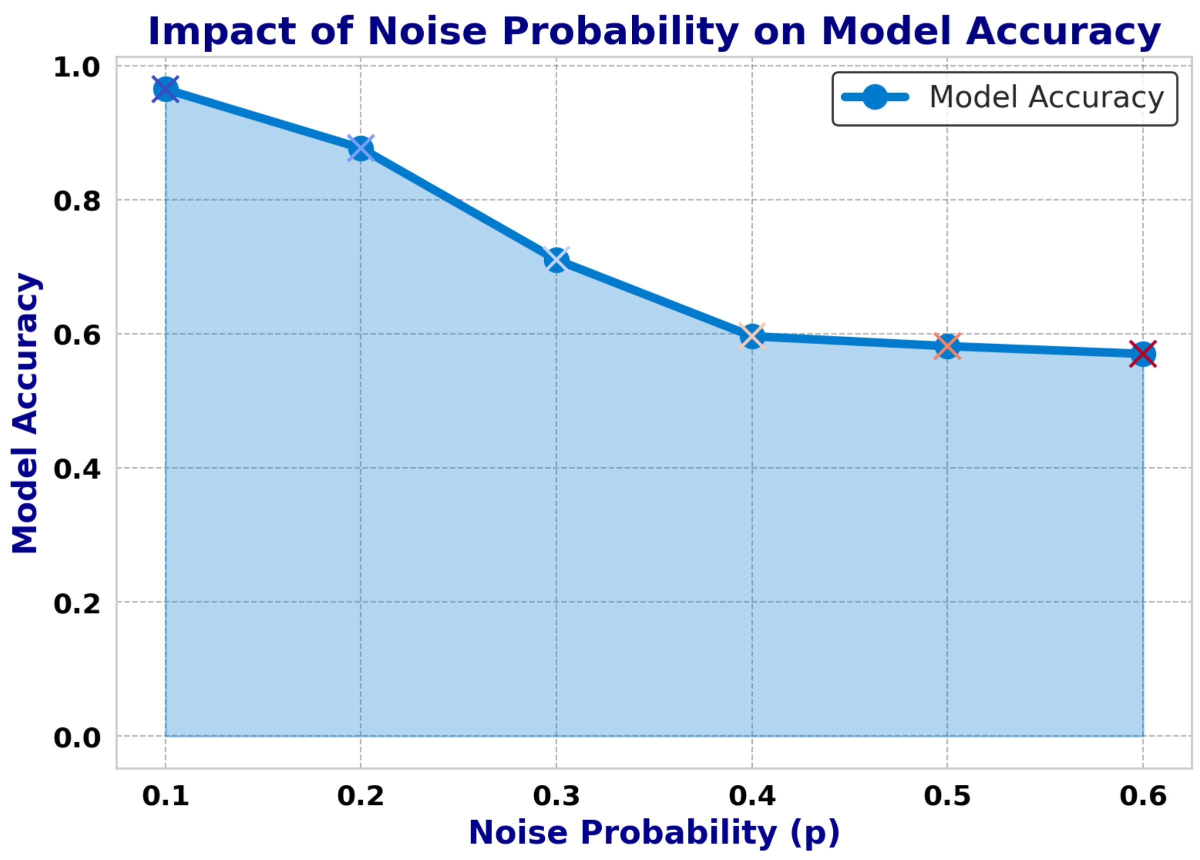


Figure 5.6: Impact of Noise Probability on Model Performance

The presence of noise in quantum computing systems significantly impacts the performance of quantum models. In this analysis, we investigate the effect of increasing noise probability (p) on model accuracy. The results indicate a clear inverse relationship between noise probability and accuracy, highlighting the sensitivity of the model to quantum noise.

As illustrated in **Figure 5.6**, the accuracy of the model declines as the noise probability increases. For $p = 0.1$, the model achieves a high accuracy of 96.49%, demonstrating robustness under minimal noise conditions. However, as p increases to 0.3, a sharp drop to 71.05% is observed, indicating that the model struggles to maintain performance under moderate noise levels. Beyond $p = 0.3$, the accuracy continues to decrease, reaching approximately 57% at $p = 0.6$, signifying the dominance of noise over signal fidelity.

The steep decline in accuracy suggests that quantum noise disrupts qubit coherence and introduces computational errors that the model cannot effectively correct. The stabilization of accuracy at higher noise probabilities indicates that beyond a certain threshold, the model reaches its noise tolerance limit, leading to consistently poor performance.

To mitigate this issue, quantum error correction techniques such as repetition encoding can be employed. Repetition codes involve encoding quantum information redundantly and applying majority voting for error correction. This approach can effectively counteract bit-flip and phase-flip noise, allowing the model to sustain higher accuracy levels even in the presence of noise.

This study demonstrates the effectiveness of the proposed Quantum-KNN (Q-KNN) model, showing superior classification performance compared to the classical KNN across multiple datasets. The results confirm that Q-KNN consistently outperforms or matches classical KNN in accuracy, F1-score, and AUC, particularly in complex datasets such as Breast Cancer and Bank Note authentication. These improvements are attributed to the quantum advantage, leveraging superposition and entanglement to enhance pattern recognition.

Additionally, the impact of quantum noise on model accuracy was analyzed, revealing a strong inverse correlation between noise probability and model performance. As noise increases, accuracy degrades significantly, highlighting the importance of quantum error correction techniques such as repetition encoding to mitigate bit-flip and phase-flip noise.

The findings of this research suggest that Q-KNN is a promising alternative for machine learning applications in high-stakes domains, such as biomedical diagnostics and financial fraud detection. Future work should explore scalability, robustness against more complex datasets, and implementation on real quantum hardware to further validate the model's practical viability.

Chapter 6

Conclusion and Suggestions for Future Work

6.1 Conclusion

This research investigated the effectiveness of Quantum-KNN (Q-KNN) compared to Classical-KNN (C-KNN) across multiple datasets, including Breast Cancer, Iris, and Bank Note Authentication. The results demonstrated that Q-KNN consistently achieved higher or equivalent accuracy than classical KNN, highlighting its potential advantages in classification tasks. In particular, Q-KNN achieved superior accuracy for complex datasets such as Breast Cancer and Bank Note, where classical KNN exhibited limitations.

The analysis of performance metrics, including accuracy, precision, recall, F1-score, and AUC, confirmed that Q-KNN maintains a strong balance between precision and recall, reducing misclassification rates. The confusion matrices further illustrated the effectiveness of Q-KNN in correctly classifying instances, reinforcing its reliability as a classification model.

Additionally, the impact of quantum noise on the model's performance was examined. It was observed that as noise probability increased, the classification accuracy of Q-KNN declined significantly. This inverse relationship between noise and model accuracy emphasizes the sensitivity of quantum machine learning models to environmental noise, which remains a critical challenge for practical implementation on quantum hardware.

The findings of this study suggest that Q-KNN offers a promising alternative to classical KNN for machine learning applications, particularly in high-stakes domains such as biomedical diagnostics and financial fraud detection. The ability of Q-KNN to harness quantum properties such as superposition and entanglement allows for enhanced pattern recognition and improved classification performance.

6.2 Suggestions for Future Work

While this study has demonstrated the advantages of Q-KNN, several areas remain open for further exploration. The following directions are proposed for future research:

- **Scalability to Larger Datasets:** The current study focused on relatively small datasets. Future work should evaluate Q-KNN on larger and more complex datasets, testing its ability to handle high-dimensional feature spaces and large-scale classification tasks.
- **Implementation on Real Quantum Hardware:** This study utilized a quantum simulator for evaluating Q-KNN. Future research should explore its performance on real quantum hardware, such as IBM Quantum or Rigetti, to assess its feasibility in practical applications.
- **Quantum Error Mitigation and Correction:** Since quantum noise significantly affects model accuracy, future work should incorporate quantum error correction techniques such as repetition encoding, quantum error correction codes, or quantum machine learning noise-adaptive strategies to enhance Q-KNN's robustness.
- **Comparison with Advanced Machine Learning Models:** While this study compared Q-KNN to classical KNN, future research should evaluate Q-KNN against other state-of-the-art classifiers, including deep learning models such as convolutional neural networks (CNNs) and transformer-based architectures.
- **Hybrid Quantum-Classical Learning Approaches:** Future research can investigate hybrid approaches that integrate quantum computing with classical deep learning techniques to leverage the best of both worlds for complex classification problems.
- **Optimization of Quantum Circuit Design:** The current quantum circuit used a basic variational approach. Further research can explore more efficient quantum circuit architectures, optimized gate arrangements, and feature embedding techniques to enhance Q-KNN's classification performance.
- **Exploring Quantum Advantage in Real-World Applications:** While this study demonstrated the theoretical benefits of Q-KNN, its practical impact needs to be assessed in real-world applications such as medical imaging diagnostics, cybersecurity, and fraud detection.

Overall, this research contributes to the growing field of quantum machine learning by demonstrating the advantages of Q-KNN over classical KNN and highlighting the challenges associated with quantum noise. As quantum hardware continues to advance, further research in this area will be crucial for unlocking the full potential of quantum-enhanced machine learning models.

References

[1] Michael A. Nielsen and Isaac L. Chuang. *Quantum Computation and Quantum Information*. Cambridge University Press, 10th anniversary edition edition, 2010.

[2] John Preskill. *Quantum Computing Lecture Notes*. 2018. Available online at http://theory.caltech.edu/~preskill/ph219/.

[3] Michael A Nielsen and Isaac L Chuang. *Quantum Computation and Quantum Information*. Cambridge university press, 2000.

[4] Frank Arute et al. Quantum supremacy using a programmable superconducting processor. *Nature*, 574(7779):505–510, 2019.

[5] Thaddeus D Ladd, Fedor Jelezko, Raymond Laflamme, Yasunobu Nakamura, Christopher Monroe, and Jeremy L O'Brien. Quantum computing with optically interfaced solid-state qubits. *Nature*, 464(7285):45–53, 2010.

[6] Maria Schuld, Ilya Sinayskiy, and Francesco Petruccione. An introduction to quantum machine learning. *Contemporary Physics*, 56(2):172–185, October 2014.

[7] Maria Schuld and Francesco Petruccione. Supervised learning with quantum computers. 2018.

[8] Seth Lloyd, Masoud Mohseni, and Patrick Rebentrost. Quantum principal component analysis. *Nature Physics*, 10:631 – 633, 2013.

[9] Aram Wettroth Harrow, Avinatan Hassidim, and Seth Lloyd. Quantum algorithm for linear systems of equations. *Physical review letters*, 103 15:150502, 2008.

[10] María Cerezo, Andrew Arrasmith, Ryan Babbush, Simon C. Benjamin, Suguru Endo, Keisuke Fujii, Jarrod R. McClean, Kosuke Mitarai, Xiao Yuan, Lukasz Cincio, and Patrick J. Coles. Variational quantum algorithms. *Nature Reviews Physics*, 3:625 – 644, 2020.

[11] Edward Farhi and Hartmut Neven. Classification with quantum neural networks on near term processors. *arXiv: Quantum Physics*, 2018.

[12] Alberto Peruzzo, Jarrod R. McClean, Peter J. Shadbolt, Man-Hong Yung, Xiaoqi Zhou, Peter J. Love, Alán Aspuru-Guzik, and Jeremy Lloyd O'Brien. A variational eigenvalue solver on a photonic quantum processor. *Nature Communications*, 5, 2013.

[13] Vojtěch Havlíek, Antonio D. Córcoles, Kristan Temme, Aram Wettroth Harrow, Abhinav Kandala, Jerry M. Chow, and Jay M. Gambetta. Supervised learning with quantum-enhanced feature spaces. *Nature*, 567:209 – 212, 2018.

[14] Maria Schuld and Nathan Killoran. Quantum machine learning in feature hilbert spaces. *Physical review letters*, 122 4:040504, 2018.

[15] Marcello Benedetti, John Realpe-Gómez, and Alejandro Perdomo-Ortiz. Quantum-assisted helmholtz machines: A quantum–classical deep learning framework for industrial datasets in near-term devices. *Quantum Science and Technology*, 3, 2017.

[16] Li-Zhen Gao, Chun-Yue Lu, Gongde Guo, Xin Zhang, and Song Lin. Quantum k-nearest neighbors classification algorithm based on mahalanobis distance. In *Frontiers of Physics*, 2022.

[17] Congcong Feng, Bo Zhao, Xin Zhou, Xiao Ting. Ding, and Zheng Shan. An enhanced quantum k-nearest neighbor classification algorithm based on polar distance. *Entropy*, 25, 2023.

[18] Jiaye Li, Jian Zhang, Jilian Zhang, and Shichao Zhang. Quantum knn classification with k value selection and neighbor selection. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 43:1332–1345, 2024.

[19] Yue Ruan, Xiling Xue, Heng Liu, Jianing Tan, and Xi Li. Quantum algorithm for k-nearest neighbors classification based on the metric of hamming distance. *International Journal of Theoretical Physics*, 56:3496 – 3507, 2017.

[20] Ziyan Tian and Sanjeev Baskiyar. Fake news detection: An application of quantum k-nearest neighbors. *2021 IEEE Symposium Series on Computational Intelligence (SSCI)*, pages 1–6, 2021.

[21] Areli-Yesareth Guerrero-Estrada, L F Quezada, and Guo-Hua Sun. Benchmarking quantum versions of the knn algorithm with a metric based on amplitude-encoded features. *Scientific Reports*, 14, 2024.

[22] Richard P Feynman. Simulating physics with computers. *International Journal of Theoretical Physics*, 21:467–488, 1982.

[23] Yuri I Manin. Computable and uncomputable. *Soviet Radio*, 1980.

[24] David Deutsch. Quantum theory, the church-turing principle and the universal quantum computer. *Proceedings of the Royal Society A*, 400(1818):97–117, 1985.

[25] Peter W Shor. Algorithms for quantum computation: discrete logarithms and factoring. pages 124–134, 1994.

[26] Isaac L Chuang, Neil Gershenfeld, and Michael Kubinec. Experimental implementation of fast quantum searching. *Physical Review Letters*, 80(15):3408, 1998.

[27] Lieven MK Vandersypen et al. Experimental realization of shor's quantum factoring algorithm using nuclear magnetic resonance. *Nature*, 414(6866):883–887, 2001.

[28] Mark W Johnson et al. Quantum annealing with manufactured spins. *Nature*, 473(7346):194–198, 2011.

[29] John Preskill. Quantum computing in the nisq era and beyond. *Quantum*, 2:79, 2018.

[30] Jacob Biamonte et al. Quantum machine learning. *Nature*, 549(7671):195–202, 2017.

[31] Edward Farhi, Jeffrey Goldstone, and Sam Gutmann. A quantum approximate optimization algorithm. *arXiv: Quantum Physics*, 2014.

[32] Peter W. Shor. Algorithms for quantum computation: discrete logarithms and factoring. *Proceedings 35th Annual Symposium on Foundations of Computer Science*, pages 124–134, 1994.

[33] Michael A. Nielsen and Isaac L. Chuang. *Quantum Computation and Quantum Information*. Cambridge University Press, 2000.

[34] Charles H. Bennett and Gilles Brassard. Quantum cryptography: Public key distribution and coin tossing. *Theor. Comput. Sci.*, 560:7–11, 2014.

[35] Richard Phillips Feynman. Simulating physics with computers. *International Journal of Theoretical Physics*, 21:467–488, 1999.

[36] Mohammad H. S. Amin, Evgeny Andriyash, Jason Tyler Rolfe, Bohdan Kulchytskyy, and Roger G. Melko. Quantum boltzmann machine. *arXiv: Quantum Physics*, 2016.

[37] Stephanie Wehner, David Elkouss, and Ronald Hanson. Quantum internet: A vision for the road ahead. *Science*, 362, 2018.

[38] Igor L. Markov. Limits on fundamental limits to computation. *Nature*, 512:147–154, 2014.

[39] C.L. Degen, F. Reinhard, and P. Cappellaro. Quantum sensing. *Reviews of Modern Physics*, 89(3), July 2017.

[40] John Preskill. *Lecture Notes for Physics 229: Quantum Computation*. 1998. http://www.theory.caltech.edu/~preskill/ph229/.

[41] R. Shankar. *Principles of Quantum Mechanics*. Springer, 2nd edition, 2012.

[42] Asher Peres. *Quantum Theory: Concepts and Methods*. Springer Science & Business Media, 2006.

[43] Dianhao Liu. Principles and characteristics of quantum computing. *Journal of Physics: Conference Series*, 2014(1):012012, sep 2021.

[44] David P. DiVincenzo. The physical implementation of quantum computation. *Fortschritte der Physik*, 48(9–11):771–783, September 2000.

[45] Nicolas Gisin, Grégoire Ribordy, Wolfgang Tittel, and Hugo Zbinden. Quantum cryptography. *Reviews of modern physics*, 74(1):145, 2002.

[46] Daniel Loss and David P DiVincenzo. Quantum computation with quantum dots. *Physical Review A*, 57(1):120, 1998.

[47] Ronald Hanson, Leo P Kouwenhoven, Jason R Petta, Seigo Tarucha, and Lieven MK Vandersypen. Spins in few-electron quantum dots. *Reviews of modern physics*, 79(4):1217, 2007.

[48] John Clarke and Frank K Wilhelm. Superconducting quantum bits. *Nature*, 453(7198):1031–1042, 2008.

[49] Rainer Blatt and David Wineland. Entangled states of trapped atomic ions. *Nature*, 453(7198):1008–1015, 2008.

[50] J. S. Bell. On the einstein podolsky rosen paradox. *Physics Physique* , 1:195–200, 1964.

[51] J. F. Clauser, M. A. Horne, A. Shimony, and R. A. Holt. Proposed experiment to test local hidden-variable theories. *Physical Review Letters*, 23:880–884, 1969.

[52] P. Grangier A. Aspect and G. Roger. Experimental realization of einstein-podolsky-rosen-bohm gedankenexperiment: A new violation of bell's inequalities. *Physical Review Letters*, 49:91–94, 1982.

[53] B. Hensen et al. Loophole-free bell inequality violation using electron spins separated by 1.3 kilometers. *Nature*, 526:682–686, 2015.

[54] A. K. Ekert. Quantum cryptography based on bell's theorem. *Physical Review Letters*, 67:661–663, 1991.

[55] Erwin Schrödinger. Die gegenwärtige situation in der quantenmechanik. *Naturwissenschaften*, 23:807–812, 1935.

[56] Markus Arndt, Olaf Nairz, Julian Vos-Andreae, Claudia Keller, Gerbrand van der Zouw, and Anton Zeilinger. Wave–particle duality of c60 molecules. *Nature*, 401:680–682, 1999.

[57] Yaakov Y. Fein, Philipp Geyer, Patrick Zwick, Filip Kiałka, Sebastian Pedalino, Marcel Mayor, Stefan Gerlich, and Markus Arndt. Quantum superposition of molecules beyond 25 kda. *Nature Physics*, pages 1–4, 2019.

[58] Christopher R. Monroe, D. M. Meekhof, B. E. King, and David J. Wineland. A "schrödinger cat" superposition state of an atom. *Science*, 272:1131 – 1136, 1996.

[59] Ryszard Horodecki, Paweł Horodecki, Michał Horodecki, and Karol Horodecki. Quantum entanglement. *Reviews of Modern Physics*, 81(2):865–942, June 2009.

[60] Michael A. Nielsen and Isaac L. Chuang. *Quantum Computation and Quantum Information*. Cambridge University Press, 2000.

[61] W. K. Wootters and W. H. Zurek. A single quantum cannot be cloned. *Nature*, 299(5886):802–803, 1982.

[62] Jacob D. Biamonte, Peter Wittek, Nicola Pancotti, Patrick Rebentrost, Nathan Wiebe, and Seth Lloyd. Quantum machine learning. *Nature*, 549:195–202, 2016.

[63] Tadashi Kadowaki and Hidetoshi Nishimori. Quantum annealing in the transverse ising model. *Physical Review E*, 58:5355–5363, 1998.

[64] Yudong Cao, Jonathan Romero, Jonathan Olson, Matthias Degroote, Peter D. Johnson, Mária Kieferová, Ian D. Kivlichan, Tim Menke, Borja Peropadre, Nicolas P. D. Sawaya, Sukin Sim, Libor Veis, and Alán Aspuru-Guzik. Quantum chemistry in the age of quantum computing. *Chemical reviews*, 2018.

[65] Sam McArdle, Suguru Endo, Alán Aspuru-Guzik, Simon C. Benjamin, and Xiao Yuan. Quantum computational chemistry. *Reviews of Modern Physics*, 2018.

[66] Alán Aspuru-Guzik, Anthony D. Dutoi, Peter J. Love, and Martin Head-Gordon. Simulated quantum computation of molecular energies. *Science*, 309:1704 – 1707, 2005.

[67] R. A. Fisher. The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, 7(2):179–188, 1936.

[68] W. H. Wolberg and O. L. Mangasarian. Multisurface method of pattern separation for medical diagnosis applied to breast cytology. *Proceedings of the National Academy of Sciences*, 87(23):9193–9196, 1990.

[69] A. Asuncion and D. J. Newman. Uci machine learning repository: Banknote authentication dataset, 2012. Available at: https://archive.ics.uci.edu/ml/datasets/banknote+authentication.

[70] H. Liu and L. Yu. Chi2: Feature selection and discretization of numeric attributes. *IEEE Transactions on Knowledge and Data Engineering*, 17(6):712–716, 2005.

[71] I. Guyon and A. Elisseeff. An introduction to variable and feature selection. *Journal of Machine Learning Research*, 3:1157–1182, 2003.

Appendix A

Codes

A.1 Classical Bit vs Qubit Figure Generation Code

We use this Python code to generate figure for Classical and Quantum Bit Comparison. . .

```
# -*- coding: utf-8 -*-
"""Thesis_figure.ipynb

Automatically generated by Colab.

Original file is located at
    https://colab.research.google.com/drive/1eBl-Q_OuaJy9Q0kjcCjwLiSIoTYSgWnr
"""

import numpy as np
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D

def plot_qubit_vs_classical():
    fig = plt.figure(figsize=(12, 6))

    # Classical Bit Representation
    ax1 = fig.add_subplot(121)
    ax1.set_title("Classical Bit Representation", fontsize=14, fontweight='bold')
    ax1.scatter(0, 1, s=600, c='blue', label=r'$0$', edgecolors='black', linewidth=1.2)
    ax1.scatter(0, -1, s=600, c='red', label=r'$1$', edgecolors='black', linewidth=1.2)
    ax1.legend()
    ax1.axis('off')

    # Quantum Qubit Representation (Simplified Bloch Sphere)
    ax2 = fig.add_subplot(122, projection='3d')
    ax2.set_title("Quantum Qubit Representation", fontsize=14, fontweight='bold')

    # Draw a semi-transparent Bloch sphere
    u = np.linspace(0, 2 * np.pi, 50)
    v = np.linspace(0, np.pi, 25)
    x = np.outer(np.cos(u), np.sin(v))
    y = np.outer(np.sin(u), np.sin(v))
    z = np.outer(np.ones(np.size(u)), np.cos(v))
    ax2.plot_wireframe(x, y, z, color='gray', alpha=0.3, linewidth=0.5)

    ax2.set_xticks([])
    ax2.set_yticks([])
    ax2.set_zticks([])
    ax2.grid(False)

    # Bloch sphere basis states
    ax2.text(0, 0, 1.1, r"$|0\rangle$", fontsize=12, color='black', ha='center', fontweight='bold')
    ax2.text(0, 0, -1.1, r"$|1\rangle$", fontsize=12, color='black', ha='center', fontweight='bold')

    # Highlight equator with a simple circle
    phi = np.linspace(0, 2 * np.pi, 100)
    ax2.plot(np.cos(phi), np.sin(phi), 0, color='black', linestyle='dashed', linewidth=1)

    # Add coordinate axes
    ax2.quiver(0, 0, 0, 1, 0, 0, color='red', linewidth=2)
    ax2.quiver(0, 0, 0, 0, 1, 0, color='green', linewidth=2)
    ax2.quiver(0, 0, 0, 0, 0, 1, color='blue', linewidth=2)

    # Qubit state vector
    theta, phi = np.pi / 4, np.pi / 3  # Example state
    x_q = np.sin(theta) * np.cos(phi)
    y_q = np.sin(theta) * np.sin(phi)
    z_q = np.cos(theta)
    ax2.quiver(0, 0, 0, x_q, y_q, z_q, color='purple', linewidth=3, label='Qubit State')
    ax2.legend()

    plt.tight_layout()
    plt.show()

# Generate the Classical vs Quantum Qubit representation
plot_qubit_vs_classical()
```

A.2 KNN Methodology

We use this Python code to describe the KNN methodology. . .

```
# -*- coding: utf-8 -*-
"""knn.ipynb

Automatically generated by Colab.

Original file is located at
    https://colab.research.google.com/drive/1Al1J4Fdv0bR6idTwTYUcylYTPJisz0Qe
"""

import numpy as np
import matplotlib.pyplot as plt
from sklearn.neighbors import KNeighborsClassifier

# Generate random data points for two classes (red and blue)
np.random.seed(42)
class_1 = np.random.randn(50, 2) + [2, 2]  # Class 1 (red)
class_2 = np.random.randn(50, 2) + [7, 7]  # Class 2 (blue)
```

```

# Create a scatter plot
X = np.vstack([class_1, class_2])
y = np.array([0] * 50 + [1] * 50)  # Labels: 0 for class_1, 1 for class_2

# Define the query point
query_point = np.array([[5, 5]])

# KNN model with K=3
knn = KNeighborsClassifier(n_neighbors=3)
knn.fit(X, y)
prediction = knn.predict(query_point)

# Create a visually appealing figure with improved style
plt.figure(figsize=(8, 6))

# Scatter plot for Class 1 and Class 2 with alpha for transparency
plt.scatter(class_1[:, 0], class_1[:, 1], c='coral', label='Class 1', edgecolors='black', s
    =100, alpha=0.7)
plt.scatter(class_2[:, 0], class_2[:, 1], c='royalblue', label='Class 2', edgecolors='black',
    s=100, alpha=0.7)

# Highlight the query point with a larger size and different color
plt.scatter(query_point[:, 0], query_point[:, 1], c='black', label='Query Point', s=200,
    marker='X', edgecolors='white')

# Highlight the K nearest neighbors with larger markers and different color
distances, indices = knn.kneighbors(query_point)
for i in indices[0]:
    plt.plot([X[i, 0], query_point[0, 0]], [X[i, 1], query_point[0, 1]], 'k--', alpha=0.5, lw
    =2)
    plt.scatter(X[i, 0], X[i, 1], c='yellow', edgecolors='black', s=150, label=f'Neighbor {i
    +1}' if i == indices[0][0] else "", alpha=0.6)

# Show the predicted label for the query point
prediction_label = 'Class 1' if prediction[0] == 0 else 'Class 2'
plt.title(f'KNN Classification with K=3\nPredicted Class: {prediction_label}', fontsize=16,
    fontweight='bold')

# Set plot limits, labels, and customize the grid
plt.xlim(0, 10)
plt.ylim(0, 10)
plt.xlabel('Feature 1', fontsize=12)
plt.ylabel('Feature 2', fontsize=12)
plt.xticks(fontsize=10)
plt.yticks(fontsize=10)
plt.grid(True, linestyle='--', alpha=0.5)

# Adding the legend with a cleaner layout
plt.legend(loc='best', fontsize=12)

# Display the plot
plt.tight_layout()
plt.show()
```

A.3 Quantum Entanglemet Figure Generation Code

We use this Python code to generate Quantum Entanglement figure. . . .

```python
# -*- coding: utf-8 -*-
"""entanglement_thesis.ipynb

Automatically generated by Colab.

Original file is located at
    https://colab.research.google.com/drive/15Qpc0cO9o9SvRemNTexlZXWbMxNgZmlH
"""

import matplotlib.pyplot as plt
import numpy as np

def plot_entanglement():
    fig, ax = plt.subplots(figsize=(6, 4))
    ax.set_xlim(-1.5, 1.5)
    ax.set_ylim(-1, 1)
    ax.set_xticks([])
    ax.set_yticks([])
    ax.set_frame_on(False)

    # Particles
    ax.scatter([-1, 1], [0, 0], s=300, color='blue', label='Entangled Particles')

    # Labels for particles
    ax.text(-1, 0.1, 'Particle A', fontsize=12, ha='center')
    ax.text(1, 0.1, 'Particle B', fontsize=12, ha='center')

    # Wavy line for entanglement
    x = np.linspace(-1, 1, 100)
    y = 0.2 * np.sin(10 * x)
    ax.plot(x, y, linestyle='dashed', color='black', label='Quantum Correlation')

    # Measurement Arrows
    ax.arrow(-1, -0.2, 0, -0.3, head_width=0.1, head_length=0.1, fc='red', ec='red')
    ax.arrow(1, -0.2, 0, -0.3, head_width=0.1, head_length=0.1, fc='red', ec='red')

    # Spin indicators
    ax.text(-1, -0.6, '   ', fontsize=14, color='black', ha='center')
    ax.text(1, -0.6, '   ', fontsize=14, color='black', ha='center')

    # Annotation for instant correlation
    ax.text(0, 0.4, 'Instantaneous Correlation', fontsize=10, ha='center', bbox=dict(facecolor
    ='white', alpha=0.8))

    ax.legend()
    plt.show()

plot_entanglement()

import matplotlib.pyplot as plt
import numpy as np

def plot_entanglement():
    fig, ax = plt.subplots(figsize=(8, 5))
    ax.set_xlim(-2, 2)
    ax.set_ylim(-1.5, 1.5)
    ax.set_xticks([])
    ax.set_yticks([])
    ax.set_frame_on(False)

    # Particles
    ax.scatter([-1.5, 1.5], [0, 0], s=500, color='blue', label='Entangled Particles')

    # Labels for particles
    ax.text(-1.5, 0.2, 'Particle A', fontsize=14, ha='center', fontweight='bold')
    ax.text(1.5, 0.2, 'Particle B', fontsize=14, ha='center', fontweight='bold')

    # Wavy line for entanglement
    x = np.linspace(-1.5, 1.5, 100)
    y = 0.3 * np.sin(10 * x)
    ax.plot(x, y, linestyle='dashed', linewidth=2, color='purple', label='Quantum Correlation'
    )

    # Measurement Arrows
    ax.arrow(-1.5, -0.4, 0, -0.5, head_width=0.15, head_length=0.15, fc='red', ec='red',
    linewidth=2)
    ax.arrow(1.5, -0.4, 0, -0.5, head_width=0.15, head_length=0.15, fc='red', ec='red',
    linewidth=2)

    # Spin indicators
    ax.text(-1.5, -1, '     Spin-Up', fontsize=14, color='black', ha='center', bbox=dict(
    facecolor='white', edgecolor='black', boxstyle='round,pad=0.3'))
    ax.text(1.5, -1, '     Spin-Down', fontsize=14, color='black', ha='center', bbox=dict(
    facecolor='white', edgecolor='black', boxstyle='round,pad=0.3'))

    # Annotation for instant correlation
    ax.text(0, 0.6, 'Instantaneous Correlation (Non-locality)', fontsize=12, ha='center',
    fontweight='bold', bbox=dict(facecolor='yellow', alpha=0.8))

    ax.legend()
    plt.show()

plot_entanglement()
```

A.4 Quantum Feature Space Figure Generation

We use this Python code to generate Quantum Feature Space Figure....

```
# -*- coding: utf-8 -*-
"""Classical_vs_Hilbert_space.ipynb

Automatically generated by Colab.

Original file is located at
    https://colab.research.google.com/drive/1OJRu0kKXSD__as0-nFk3RXv0s_zIDREu
"""

import numpy as np
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D

```

```python
def plot_classical_vs_quantum():
    fig = plt.figure(figsize=(14, 6))

    # Classical Space - Poor Class Separation
    ax1 = fig.add_subplot(121, projection='3d')
    class_1_x = np.random.uniform(-1, 0, 50)
    class_1_y = np.random.uniform(-1, 0, 50)
    class_1_z = np.random.uniform(-0.1, 0.1, 50)
    class_2_x = np.random.uniform(0, 1, 50)
    class_2_y = np.random.uniform(0, 1, 50)
    class_2_z = np.random.uniform(-0.1, 0.1, 50)

    ax1.scatter(class_1_x, class_1_y, class_1_z, c='blue', label='Class 1', alpha=0.7,
    edgecolors='k')
    ax1.scatter(class_2_x, class_2_y, class_2_z, c='red', label='Class 2', alpha=0.7,
    edgecolors='k')
    ax1.set_title("Classical Feature Space", fontsize=14)
    ax1.set_xlabel("X-axis", fontsize=12)
    ax1.set_ylabel("Y-axis", fontsize=12)
    ax1.set_zlabel("Z-axis", fontsize=12, labelpad=10)
    ax1.legend()

    # Quantum Feature Space - Better Separation
    ax2 = fig.add_subplot(122, projection='3d')
    theta_class_1 = np.random.uniform(0, np.pi / 3, 50)
    phi_class_1 = np.random.uniform(0, np.pi, 50)
    x_class_1 = np.sin(theta_class_1) * np.cos(phi_class_1)
    y_class_1 = np.sin(theta_class_1) * np.sin(phi_class_1)
    z_class_1 = np.cos(theta_class_1)

    theta_class_2 = np.random.uniform(2 * np.pi / 3, np.pi, 50)
    phi_class_2 = np.random.uniform(np.pi, 2 * np.pi, 50)
    x_class_2 = np.sin(theta_class_2) * np.cos(phi_class_2)
    y_class_2 = np.sin(theta_class_2) * np.sin(phi_class_2)
    z_class_2 = np.cos(theta_class_2)

    ax2.scatter(x_class_1, y_class_1, z_class_1, c='blue', label='Class 1', alpha=0.7,
    edgecolors='k')
    ax2.scatter(x_class_2, y_class_2, z_class_2, c='red', label='Class 2', alpha=0.7,
    edgecolors='k')
    ax2.set_title("Quantum Feature Space", fontsize=14)
    ax2.set_xlabel("X-axis", fontsize=12)
    ax2.set_ylabel("Y-axis", fontsize=12)
    ax2.set_zlabel("Z-axis", fontsize=12, labelpad=10)
    ax2.legend()

    plt.tight_layout()
    plt.show()

plot_classical_vs_quantum()
```

A.5 Quantum KNN accuracy graph generation code

We use this Python code to generate the accuracy graph between Proposed Q-KNN and KNN. . . .

```python
# -*- coding: utf-8 -*-
```

```python
"""accuracy_graph.ipynb

Automatically generated by Colab.

Original file is located at
    https://colab.research.google.com/drive/1zMrSqXTBJZ6rweDhNiel-5XG8OOv6EMt
"""

import matplotlib.pyplot as plt
import numpy as np
import seaborn as sns

# Data
datasets = ['Breast Cancer', 'Iris', 'Bank Note']
qknn_accuracy = [0.9825, 1.0000, 0.9927]
knn_accuracy = [0.9298, 1.0000, 0.9855]
x = np.arange(len(datasets))  # Label positions

# Set advanced academic-style aesthetics
sns.set_style("whitegrid")
sns.set_context("notebook")

# Create figure and axis
plt.figure(figsize=(10,6))
bar_width = 0.4

# Using gradient colors for a visually appealing effect
colors = sns.color_palette("coolwarm", n_colors=2)

# Plot bars with gradient-style shading and enhanced visualization
plt.bar(x - bar_width/2, qknn_accuracy, bar_width, label='Proposed Q-KNN Accuracy',
        color=colors[0], edgecolor='black', alpha=0.85)
plt.bar(x + bar_width/2, knn_accuracy, bar_width, label='KNN Accuracy',
        color=colors[1], edgecolor='black', alpha=0.85)

# Adding value labels to bars
for i in range(len(datasets)):
    plt.text(x[i] - bar_width/2, qknn_accuracy[i] + 0.005, f"{qknn_accuracy[i]:.3f}",
             ha='center', fontsize=12, fontweight='bold', color='black')
    plt.text(x[i] + bar_width/2, knn_accuracy[i] + 0.005, f"{knn_accuracy[i]:.3f}",
             ha='center', fontsize=12, fontweight='bold', color='black')

# Labels and title with an academic research style
plt.xlabel("Dataset", fontsize=14, fontweight='bold')
plt.ylabel("Accuracy", fontsize=14, fontweight='bold')
plt.title("Comparative Performance of Q-KNN and KNN", fontsize=16, fontweight='bold')
plt.xticks(x, datasets, fontsize=12, fontweight='bold')
plt.yticks(fontsize=12)
plt.ylim(0.9, 1.05)  # Set limits to enhance visualization
plt.legend(fontsize=12, frameon=True, fancybox=True, shadow=True, loc='upper right')
plt.grid(axis='y', linestyle='--', alpha=0.6)

# Show plot
plt.show()
```

B.Sc.
Engg.
EEE
BUET

A Novel Quantum KNN Algorithm

Asif Akhtab Ronggon

March
2025