Evolution Of Quantum Computing In Machine Learning

Sunny Kumar[1, a)] Lokesh Mishra[1, b)] Gufran Ali [1,c)] Himanshu Sharma[1, d)]

[1]*CSE Dept., Sharda University, India*

Abstract. Quantum physics was one of the major breakthroughs in the field of physis in the 20th century which describes the behavior of particles at the quantum level including subatomic particles. Later in the 21st century procedural computing system emerged as a bottle neck for many standard computing problems including machine learning algorithms. Eventually many researchers concluded that quantum physics could be a major application in computing system that will solve the problem which cannot be solved with procedural computing system, called quantum computing. Quantum computing works on the principle of superposition of different possible states at a time which allows computing devices to solve problems more efficiently, with the help of qubits (bits in quantum computing).This paper provides a comprehensive study of history of quantum physics and its evolution over the period of time and its emergence with the standard computing system which is now called as quantum computing system. Finally, it contains the history of quantum physics, Procedural Computing system its drawback, development of quantum computing, application of quantum computing in machine learning including QSVM(quantum support vector machine),K-means clustering, quantum simulation, quantum neural networks, quantum deep learning ,quantum machine learning models and algorithms including quantum HHL algorithm, quantum SVM, quantum classifier and current work on quantum classifier.

1. INTRODUCTION

An important development in the fields of computer science and information technology is the advent of quantum computing. By using the ideas of quantum physics, quantum computing is able to do intricate computations at rates that were previously unthinkable for traditional computers. The prospects of quantum computing became apparent with the groundbreaking work of David Deutsch [1], who established the theoretical framework, and Peter Shor [2]who showed the great potential for quantum algorithms. Further studies by Lov Grover in [3] and Shor [4] further highlighted how transformational quantum computing can be. The emergence of quantum computing has the potential to transform encryption, enhance intricate simulations,Artificial Intelligence and reveal innovative resolutions for an extensive array of scientific and industrial problems. Researchers' attention has been captured by this innovative technology, which has sparked a dynamic and quickly developing sector devoted to realising its full potential. One of artificial intelligence's most exciting new frontiers is the use of quantum computing into machine learning. Machine learning techniques might be completely transformed by quantum computing's special computational powers, which allow for the processing of enormous datasets and intricate calculations at previously unthinkable rates. By proving that quantum computers can replicate physical systems—a critical skill for many machine learning applications—Seth Lloyd's groundbreaking work [5] established the theoretical groundwork for quantum machine learning. Furthermore, a perceptive summary of the possible effects of quantum computing on data mining and machine learning may be found in Peter Wittek's study [6].Combining quantum computing and machine learning might improve pattern recognition, solve optimisation issues, and make it easier to create AI systems that are more reliable and effective. With its cutting-edge methods for predictive modelling and problem-solving, this rapidly developing multidisciplinary discipline is expected to significantly enhance artificial intelligence. As quantum hardware capabilities continue to grow and algorithmic frameworks mature, the integration of quantum computing with artificial intelligence is no longer a theoretical exercise but a rapidly approaching reality. Current efforts by leading technology firms such as IBM, Google, Microsoft, and Rigetti, alongside academic research, are accelerating the development of quantum cloud platforms and open-access quantum processors. These initiatives enable researchers worldwide to experiment with quantum-enhanced machine learning models, paving the way for early proof-of-concept applications in fields such as pharmaceuticals, financial analytics, cryptography, logistics optimization, and materials science. Moreover, the interdisciplinary nature of quantum machine learning demands a fusion of expertise in quantum physics, computer science, applied mathematics, and data science. This convergence is expected to drive novel algorithms that leverage quantum principles such as superposition, entanglement, and quantum parallelism to deliver exponential speedups for select classes of problems, particularly those related to optimization, pattern recognition, and probabilistic inference. While significant challenges remain, including hardware limitations, error correction, data encoding inefficiencies, and the complexity of developing quantum-native algorithms—the potential rewards are transformative. The continued synergy between quantum computing and machine learning promises not only to redefine the computational boundaries of AI but also to unlock solutions to problems previously considered unsolvable within reasonable timeframes. This paper aims to provide a comprehensive exploration of this evolving intersection, documenting the historical context, foundational technologies, key algorithms, and emerging applications that position quantum machine learning as a pivotal force in the future of artificial intelligence and high-performance computing.

2. EVOLUTION OF QUANTUM COMPUTING

The emergence of quantum computing is an enthralling and fast evolving topic that has the potential to revolutionise computing. The notion of quantum computing was developed in the 1980s by physicist Richard Feynman. In the 1990s, Peter Shor's technique demonstrated quantum computers' capacity to effectively factor big numbers, posing a challenge to traditional encryption approaches. Quantum bits, or qubits, are at the heart of quantum computing because they may exist in numerous states at the same time (superposition) and are entangled with one another. The advancement of different qubit technologies, including as ions, superconducting circuits, and photon-based qubits, has been critical in the field's advancement.Quantum algorithms, such as Grover's algorithm for unstructured search and quantum simulation algorithms, expanded on the capabilities of quantum computers. Companies like as IBM, Google, Rigetti, and D-Wave have taken the lead in developing experimental quantum computers employing various qubit technologies throughout the years. Quantum error correction codes and noise and decoherence mitigation approaches have proven critical in making quantum computers more dependable.Google's claim of quantum supremacy in 2019 with their 53-qubit processor was a huge step forward, but scaling quantum systems to thousands of qubits and sustaining low error rates remain hurdles. Quantum networking and cloud services have recently evolved, allowing academics and enterprises to test quantum algorithms and applications. Quantum programming languages and software libraries have aided in the accessibility of quantum development. As quantum computing moves from research to commercial applications, it has the potential to disrupt businesses and solve complicated issues that are now beyond the grasp of traditional computers, making it an exciting and innovative sector.

2.1 History of Quantum Physics

It all began when scientists realised that the traditional laws of physics didn't exactly apply when dealing with incredibly tiny objects like atoms and their components. This realisation led to the development of quantum physics, which is a fascinating narrative that changed how world view the fundamental components of the world.Scientists were baffled by how light is emitted from heated things and how light may eject electrons from metals in the late 1800s. In 1900, a physicist by the name of Max Planck proposed a novel notion: energy exists as small packets or pieces. The quantum concept was born out of this [7]. Then, in 1905, Albert Einstein presented the hypothesis that light itself is composed of small energy particles, like microscopic bullets [8]. This was a significant shift from the previous view that light behaved like waves. When Niels Bohr explained how electrons travel around an atom's nucleus in 1913 [9], it was a significant milestone. He combined the ancient laws of motion with the novel quantum theories, although his model had significant issues with larger atoms.The Bohr atom model's inability to explain the Zeeman effect and the Stark effect [10]is one of its major flaws.The Bohr atom model's inability to account for the wave-particle duality [11] of electrons is another flaw. Despite having both wave-like and particle-like characteristics, electrons are exclusively treated as particles in the Bohr atom model. Later, with the help of double-slit experiment it was discovered that light posses dual nature which was performed earlier in 1801 by Thomas Young.This was one of the major breakthorugh that paved a way for the advanced quantum physics.A barrier with two slits is targeted by a laser beam during the experiment. There is a light-catching screen behind the barrier. When only one slit is exposed, the light imprints on the screen a pattern that one would anticipate from particles striking it. However, an odd phenomenon occurs when both slits are open. An interference pattern, like what would be seen when waves interact, such as when you toss two stones into a pond, shows on the screen instead of just two lines. With its peaks and troughs interacting and forming patterns, this interference pattern demonstrates how light behaves like a wave. The catch is that the interference pattern vanishes when you attempt to determine which slit the light passes through, as by positioning a detector at each slit. It appears the light "knows" it is being observed and adjusts, acting more like a particle that passes through one slit or the other. This experiment is a perplexing demonstration of light's dual nature, which allows it to act like a particle while still producing patterns like a wave. It's a crucial piece of proof for comprehending the peculiarities of quantum physics. Werner Heisenberg in his uncertainty principle [12]] asserted in 1927 that humans cannot fully understand the nature of small particles. There is a limit on just how much information we can get about objects, such as their location and rate of movement. The realization that the world wasn't as deterministic as we imagined was a huge shock. Schrodinger proposed a peculiar scenario in 1935 with a cat that, up until we peek inside the box, is both alive and dead. This served to illustrate just how bizarre some quantum concepts may be, also known as schrodinger's equation [13] [14] .It was necessary to put quantum physics to the test. Research conducted like the double-slit experiment and Bell's theorem [15] demonstrated the unusual and unexpected phenomena that arise at the lowest scales and defy common sense. We had the Standard Model, which included a lengthy list of the particles and their functions, by the second half of the 20th century. It clarified the dynamics and minuscule components that create everything.Amazing technology have also been created using these quantum concepts. For instance,quantum computers can tackle issues that traditional computers cannot that we will discuss in later sections.

2.2 Procedural Computing System and its drawbacks

Procedural commuting describes the sequence in which quantum gates are applied to qubits [16]. The sequence in which they are used can significantly affect the behavior and effectiveness of a quantum algorithm. Quantum gates [11] are the essential components of quantum circuits [17]. several procedural commuting disadvantages:

- Procedural computing systems are constrained by the speed of light because they rely on electrical signals for communication between their many components. Since these electrical impulses can only move at the speed of light, doing complicated calculations that need a lot of data to be processed may take a very long time

- Error-prone: Because procedural computing systems employ electrical signals to represent and process data, they are subject to mistakes. Rounding mistakes and system noise may both taint electrical signals.Considering that procedural computing systems are error-prone and constrained by the speed of light, they are not ideal for addressing complicated tasks.

- Speed restriction: By carrying out a series of instructions one at a time, procedural computing systems solve issues. Which means that the time that will be needed to solve an issue is directly related to the quantity of instructions that must be carried out. This may result in extremely lengthy execution times for complicated tasks.

- Error propensity: Because procedural computing systems employ electrical signals to represent and process data, they are susceptible to mistakes. Rounding mistakes and system noise may both taint electrical signals. Over time, these mistakes may compound and provide false findings. Because of these factors, procedural computing methods are not suitable for handling large problems that call for numerous calculations or that are very error-prone.

Here are list of some instances of complicated issues that procedural computing systems find hard to solve:

- The Schrödinger equation[13] a partial differential equation that explains the behavior of quantum systems[18] must be solved in order to simulate the behavior of molecules, making it a challenging task. Even the most powerful supercomputers can only model the behavior of tiny molecules when attempting to numerically solve the Schrödinger equation.
- Finding the ideal mix of components and structures to produce the desired qualities is a difficult task in the design of novel materials. It is sometimes essential to attempt several alternative combinations before finding a solution since this is a highly challenging problem to tackle using conventional techniques.

- Modeling the financial sector:Because of these factors, procedural computing methods are not suitable for handling large problems that call for numerous calculations or that are very error-prone.
- Artificial intelligence: The creation of algorithms that are able to learn from and adjust to new knowledge is a hard challenge. Traditional AI systems are frequently incapable of learning new skills or making adjustments, and they are readily deceived by hostile cases.In comparison to procedural computing systems, quantum computing systems have the potential to be more effective and precise in resolving all of these challenging issues, Because quantum computers are less error-prone and are not constrained by the speed of light.

2.3 Development of Quantum Computing

When scientist like Richard Feynman[19] recognized that the rules of quantum physics might be applied to develop computers that were far more powerful than any conventional computer, the concept of quantum computing was born.The well-known American physicist Richard Feynman significantly influenced the creation and idea of quantum computing. Feynman made the claim that it was difficult to simulate quantum systems using classical computers and that quantum computers may provide a more effective solution in the early 1980s. His well-known talks and debates on this subject paved the path for the development of quantum computing [19].The main contributions and ideas of Feynman to quantum computing are as follows:

- **Feynman's Vision**: In his well-known 1981 lecture, "Simulating Physics with Computers," [20]The difficulties that classical computers have while modeling quantum systems were highlighted in Richard Feynman's lecture "Simulating Physics with Computers," which also proposed the idea of quantum computers as a more logical and effective method of simulating and comprehending quantum physics. The research and advancements in the field of quantum computing were greatly influenced by this speech.

- **Quantum Turing Machines**: Even though it isn't stated specifically in the presentation, Feynman's idea fits inside the purported parameters of a "quantum Turing machine." A theoretical model that describes how a quantum computer might carry out calculations is known as a quantum Turing machine. It adds quantum operations and states to the traditional Turing machine concept[20].
- **Qubit**: Although Feynman did not coin the name "qubit" (quantum bit), his research helped to establish the possibility of quantum information being stored in quantum states, such as the superposition of quantum states. The foundation of quantum computing is based on this idea of qubits[20]. Richard Feynman, a physicist, made additional significant contributions when he proposed that quantum systems may mimic and resolve quantum problems more quickly than conventional computers.David Deutsch, a scientist, first introduced the idea of quantum bits, also known as qubits, as the basic building blocks of quantum information in the 1960s. Qubits may exist in a superposition of states, unlike conventional bits.

- **Quantum Algorithms**: The creation of quantum algorithms is emphasized in Feynman's presentation [20]as a way to utilize the potential of quantum computers. These algorithms accomplish some jobs noticeably quicker than classical algorithms by utilizing superposition, entanglement, and quantum parallelism. Two prominent examples of quantum algorithms that show how quantum computing may be used to solve particular problems more effectively are Shor's and Grover's algorithms.

Feynman suggests the concept of simulating other quantum systems using quantum systems themselves. He contends that if quantum computers could be created, they may offer a more accurate and effective way to simulate quantum mechanics. These hypothetical quantum computers would run on quantum bits, or qubits, which are ideal for mimicking quantum processes because they can represent superpositions and entanglement. While Richard Feynman's contributions to quantum computing were more conceptual and foundational than practical, his insights helped establish the theoretical framework for quantum computing. His ideas laid the groundwork for subsequent developments in the field, leading to the exploration of quantum algorithms and the construction of actual quantum computers.Feynman suggested that quantum systems like atoms and molecules couldn't be accurately simulated by traditional computers. This is due to the ability of quantum systems to superposition, or the simultaneous existence of several states. Since traditional computers can only handle one state at a time, simulating a quantum system would require an exponentially enormous amount of memory.Over the course of several decades, advances in physics, mathematics, and computer science came together to give rise to the concept of quantum computing.

- **Quantum Mechanics**: The early 20th century development of quantum mechanics laid the groundwork for quantum computing. Wave-particle duality, superposition, and entanglement are important ideas. Pioneering scientists like Max Planck, Niels Bohr, and Erwin Schrödinger created the mathematical foundation for quantum mechanics in the 1920s and 1930s.

- **Quantum Superposition (1926)**: In his wave equation, Erwin Schrödinger developed the idea of quantum superposition. Quantum particles, such as electrons, may exist in several states at the same time due to superposition.

- **Quantum Entanglement**: Quantum entanglement is the theory that states that the properties of two or more particles may become connected to the point that measuring one particle immediately alters the state of the other or particles, was first put out by Albert Einstein, Boris Podolsky, and Nathan Rosen (EPR). Einstein notoriously dismissed this theory as "spooky action at a distance."

- **Shor's Algorithm**: Shor's algorithm, a quantum method created by mathematician Peter Shor, showed how quantum computers might factor enormous numbers exponentially more quickly than conventional computers. Effective factoring of huge numbers is essential for cracking RSA and other traditional encryption techniques [21].

- **Grover's Algorithm**: In 1996 Lov Grover created Grover's method, which demonstrated that unstructured search jobs may be completed by quantum computers quadratically quicker than by classical computers. This technique has applications in tackling optimization issues and searching through enormous datasets [22].

- **Early quantum hardware**: from the 1980s through the 1990s Experimentalists made progress in creating the first primitive quantum devices in the 1980s and 1990s, such as ion trap and superconducting qubit-based systems. The hardware for quantum computing was built on the shoulders of these early initiatives.

- **Quantum Information Theory**: At the same time, scientists created quantum information theory, a discipline devoted to comprehending and utilizing quantum phenomena for communication and computation. Key contributors were John Preskill, Charles Bennett, and Gilles Brassard.

- **Complexity and Quantum Algorithms**: For tackling certain issues, quantum researchers investigated quantum algorithms, which frequently showed exponential speedup in comparison to classical methods. This study emphasized the potential uses of quantum computing for simulation, optimization, and cryptography.

- **Quantum Hardware Advancements**: Quantum processors from firms like IBM, Google, and Rigetti have pushed the feasibility of practical quantum computing closer to reality. A key milestone was reached when a quantum computer outperformed a traditional supercomputer at a given job (as Google showed in 2019).

- **Growing Investment and Interest**: Governments, academia, and the corporate sector have all recently shown a spike in interest in and investment in quantum computing. Building more potent and reliable quantum computers is an important area of research and development for businesses and academic institutions.

3. APPLICATIONS OF QUANTUM COMPUTING

The vast permutations that result in twice as much memory in quantum computers with each qubit added are what give quantum computation its potency. We want N bits of binary numbers in order to define a N bits classical bits system. We now understand that there are two potential definite states in quantum systems, which are 0 &1.

A bipartite quantum system's general state may be expressed as

$$\phi = \alpha|00\rangle + \beta|01\rangle + \gamma|10\rangle + \delta|11\rangle \quad \textit{Eqn. 1}$$

It is clear that a two-qubit quantum system yields four classical bits of information (α, β, γ, and). Analogously, we may obtain 2^n bits of classical information from the N-qubit quantum system more quickly than via any other conventional method [23].

3.1 K-means clustering and Nearest Neighbour Classification

It is a machine learning standard algorithm. The K-nearest neighbor (KNN) method evaluates a new data item that has to be categorized based on how similar it is and how its neighbors are categorised, taking into account all of the prior data. A vector and another vector are more similar the closer they are to each other. The inner product, the Hamming distance, or the Euclidean are common techniques for calculating proximity or separation. In [24]The writers employ an overlap or fidelity strategy of two quantum state to calculate how similar two vectors are. Through the use of a swap test subroutine, the overlap is obtained. Based on [24] the authors in [25]suggested a time-consuming quantum method, time O(log MN). This resulted in an exponential acceleration [26].Additionally, techniques for calculating the distance between feature vectors have been provided by the authors of [27]. Based on the swap test, the technique offers two ways to calculate Euclidean distance: directly and through the inner product. Grover's search is combined with the use of amplitude amplification. Qubits, on the other hand, express classical information differently.Compared to Monte Carlo techniques, the approach, in the worst case scenario, results in polynomial reductions.

3.2 Quantum SUPPORT VECTOR MACHINE (QSVM)

For data models, supervised machine learning methods like SVMs are frequently utilized. They are mostly employed in regression and classification analyses. The data model is trained using a test sample, and each value is assigned to one of the possible categories. Finding the best hyperplane to divide two-class areas precisely and serve as a decision boundary for upcoming inputs is the challenge at hand in these types of situations.Grover's search was modified in the early 2000s when the authors of [28] suggested the first iteration of the quantum SVM. More potent techniques have lately been created. The data input may originate from quantum subroutines that prepare quantum states or from sources like qRAM that accesses classical data. To be more precise, the best hyperplane and input vector testing are achieved using quantum phase estimation and matrix invasion, which in theory calls for time poly (log N) The matrix dimension N is needed to generate a quantum version of the hyperplane vector. The HHL algorithm may be used to analyze data using the techniques outlined in [29–31].

3.3 Quantum neural networks (QNNs) and deep learning

Computational neural networks, or QNNs, operate according to the laws of quantum mechanics. Particular study is being done on artificial neural networks because of their use in big data applications and pattern detection. Theories like interference, parallelism, and entanglement are thought to be useful. The concept of quantum artificial networks has been examined in an increasing number of papers [32–34].The idea behind current work in the field is to create a brain unit that is in a superposition of the active and non activated states by substituting a qubit for the traditional binary bit.Deep quantum learning networks may be easily built with quantum annealers, which are readily scalable, commercially accessible, and suitable [35].The Boltzmann machine, a deep learning network, is the most approximable [36].Quantum information in the form of qubits is produced by the quantum Boltzmann machine. In their assessment, Schuld et al. [37]came to the conclusion that no ideas fully use the potential of quantum computing. This has remained speculative up to now since normalization of quantum states is required. An incorrect state would result from using a unitary operator such as addition in a Hilbert space, which is a real or complex inner product space that satisfies specific criteria. In addition, QNN has linear dynamics while traditional neural networks have non-linear dynamics.

3.4 Hidden quantum Markov models

A stochastic model known as a Markov model is used to simulate temporal or sequential data and aid in future value prediction by using available data. A hidden Markov model (HMM) is a type of Markov model in which its states are hidden and are only visible when the state provides them as an output. Speech that has been recorded exemplifies the Markov chain of subsequent words. For modeling sequential data, HMM is very helpful in domains like Natural Language Processing (NLP).Hidden quantum Markov models (HQMMs) were initially introduced in 2010 by the authors of [38].HQMMs are superior than traditional HMMs in that they are a generalization. To implement the HQMM, the authors of [39]have suggested open quantum systems with immediate feedback. Furthermore, they point out that HQMMs may be used as stochastic process simulators. There has been a proposal for an iterative maximum likelihood method recently [40]. The algorithm could be able to better represent some sequential data and learn HQMM.

3.5 Quantum Simulation

When a classical computer attempts to replicate quantum physics, it runs into several issues. Because the huge number of explicit states of the quantum system must be stored in a massive quantity of computational memory, direct simulation on a classical computer is exceedingly difficult to do. This occurs because the number of parameters used to define quantum states grows exponentially with system size [41]. Richard Feynman first presented the concept of quantum simulators in 1982 to accomplish tasks that the most potent supercomputers or classical simulation techniques were unable to. Feynman presented an alternate approach to quantum simulation, which is characterized as using quantum mechanics to simulate a quantum system. "One controllable quantum system to simulate another" is the concept. Quantum simulators might address quantum many-body issues using this method. It would provide novel outcomes that are impossible to forecast and exceedingly difficult to model traditionally. Since they are quantum systems, they would aid in our ability to shed further light on the enormous field of quantum phenomena. It would also enable us to test and validate other models. One may study difficult problems in condensed matter physics, such quantum magnetism and correlated electrons. The fact that quantum simulation eliminates the need for explicit quantum gates is one of its primary advantages. There is less accuracy required when dealing with quantum simulations, and error correction is not a major obstacle. The fact that a quantum simulation may be executed with only a few tens of qubits is a clear benefit of simulation over other quantum computing methods. Conversely, popular algorithms like Shor's algorithm need thousands of qubits to function. Second, soon, a quantum simulator might be constructed using existing technology. Thirdly, there are many other fields in which quantum simulation finds use, including condensed-matter physics, cosmology, nuclear physics, and quantum chemistry. Interest in quantum simulators increased consequently.

3.6 Learning and renormalization process

To make sense of information it has never seen before, machine learning utilizes data to find patterns. Some features are just too complex to be represented using traditional numerical methods. This is the point at which problem-solving using machine learning techniques is crucial. Recent developments in machine learning have shown how useful it may be to use ML techniques like pattern recognition or classification to identify phases of matter or to use neural networks to approximate arbitrary functions in a nonlinear manner. With its usage in speech-to-text and text-to-speech converters, internet searches, recommendation engines, content filtering, cameras, cellphones, sentiment analysis, and a plethora of other applications, machine learning has become a staple of modern society. Deep learning, or deep neural networks, are usually required for these applications. The discovery of asymptotic freedom in quantum chromodynamics, the Kosterlitz–Thouless phase transition, and the groundbreaking research on critical events are all based on the renormalization group (RG) method. The RG approach is a conceptual framework that includes, among other methods, real-space RG, functional RG, and density matrix RG. Finding the "relevant" degrees of freedom and repeatedly integrating the "irrelevant" ones is the core of RG. Thus, we arrive at an effective theory that is low-energy and universal. The RG method integrates out the remaining degrees of freedom while systematically retaining the "slow" degrees of freedom, revealing the universal qualities that ultimately dictate their physical features. Nonetheless, determining significant degrees of freedom is the primary issue. Think of a feature map that changes any given data set X to a coarser-grain scale.

$$x \rightarrow \phi_{\lambda}(x)$$

Eqn. 2

$$f(x) = g(x) + \frac{1}{\lambda} f\left(\phi_{\lambda}(x)\right) \quad \textit{Eqn. 3}$$

In order to represent the fact that the free energy is both scale-invariant and size-extensive close to a critical point, the RG theory demands that the free energy F(x) be scaled. The Renormalization Group Equation (RGE) at its Foundation:

4. QUANTUM MACHINE LEARNING MODELS

Quantum machine learning is an interesting frontier at the confluence of quantum computing and conventional machine learning, with the potential to revolutionize a wide range of scientific and industrial disciplines. Unlike traditional computers, quantum computers use quantum physics' unique features, such as superposition and entanglement, to handle information in a fundamentally different way. This opens the door to addressing complicated optimization issues, speeding up database searches, and improving machine learning techniques. To solve conventional machine learning problems more effectively, quantum algorithms such as the quantum support vector machine (QSVM) and quantum neural networks have been developed [42]. While quantum machine learning is still in its early phases, it has the potential to greatly speed up data processing, pattern identification, and optimization tasks, with applications ranging from cryptography to drug development.

4.1 Quantum HHL Algorithm.

In 2009, Aram Harrow, Avinatan Hassidim, and Seth Lloyd proposed a quantum technique for solving linear systems of equations. More precisely, for a given linear system of equations Ax = b, the method may estimate the outcome of a scalar measurement on the solution vector b. In this context, A is a N × N Hermitian matrix with a spectral norm bounded by 1 and a finite condition number $\kappa = |\gamma_{max}/\gamma_{min}|$. Only when matrix A is well-conditioned (i.e., its singular values fall between 1/κ and 1) and sparse (at most poly (log *N*) entries per row) can the HHL method be executed effectively. The phrase "scalar measurement" is also emphasized here: the solution vector x generated by the HHL subroutine cannot be read out directly since it is (approximately) encoded in a quantum state x qubits and cannot be directly readout; at most, we may measure it in a basis or sample it using a quantum mechanical operator M, i.e. $\langle \tilde{x}|M|\tilde{x} \rangle$, to establish certain statistical properties of $|x|$ in a single algorithm run. To even identify a particular entry in the solution vector, the algorithm would need to run around N times. Moreover, the HHL requires a quantum RAM (in theory)—that is, a memory that can, from the entries $\{b_i\}$ of **b**, construct the superposition state |b} (encoded **b**) all at once without requiring parallel processing components uniquely for each entry. The HHL operates in the stated $O(\log n^2\, s^2 \kappa^2/\varepsilon)$time only if all these requirements are met, where ε is the desired error and s is the sparsity parameter of matrix A, or the maximum number of nonzero elements in a row [43, 44].At first glance, the method might not appear very helpful given all these limitations, but it's vital to grasp the context in this case. HHL is not intended to be used as a stand-alone method for solving systems of linear equations in logarithmic time; rather, it is intended to be used as a subroutine in other algorithms. Stated otherwise, the HHL can be implemented in specific situations where it is possible to prepare |b} in an efficient manner, where the unitary evolution e^{-iAt} can be applied within a realistic time frame, and when just a portion of the solution vector **x**'s observables are needed, as opposed to all its members. The 2013 study by Clader et al. [45] provides a detailed illustration of one such HHL use case with a highly practical implementation such that electromagnetic scattering cross-sections of any given target may be calculated more quickly than with a traditional approach (Fig. 1)

Phase estimation, controlled rotation, and uncomputation are the three processes that make up the HHL algorithm [46, 47]. As the algorithm's initial step, let $A = \sum_j \lambda_j |u_j\rangle\langle u_j|$ Considering the scenario $|b\rangle = |u_j\rangle$, is one of the eigenvectors of **A** as the input state. The following mapping may be implemented using the quantum phase estimation approach given a unitary operator U with eigenstates ~ $\langle u_j\rangle$ and associated complex eigen values $e^{i\varphi j}$:

$$|0 \mid u_j\rangle \rightarrow |\tilde{\phi}\rangle u_j \qquad \textit{Ean. 4}$$

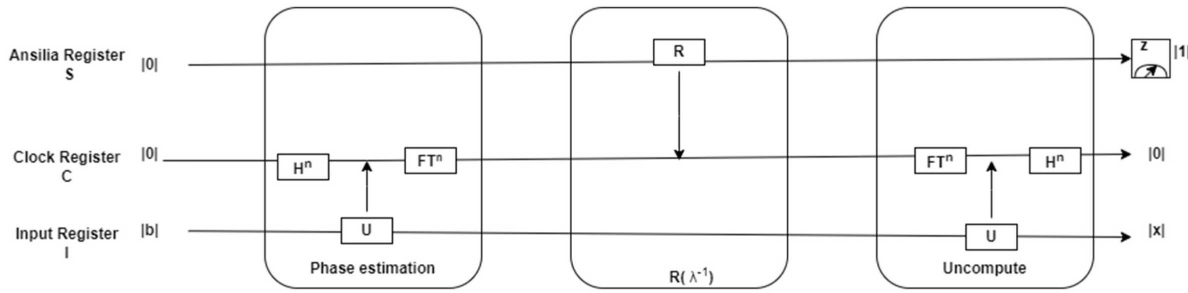

Fig. 1 HHL algorithm schematic

Here, $\tilde{\phi}$ is a precise binary representation of ϕ. When dealing with a Hermitian matrix A that has eigenstates $\{u_j\}$ and matching eigenvalues u_j, the matrix e^{iAt} is unitary, with eigenvalues $e^{i\lambda jt}$ and eigenstates $\langle u_j\rangle$. Thus, to apply the mapping, the phase estimation approach may be used to the matrix e^{iAt}.

$$|0\rangle|u_j\rangle \rightarrow \tilde{\lambda}_j\rangle|u_j \qquad \textit{Eqn. 5}$$

where $\tilde{\lambda}_j$ is the binary representation of λ_j.

The algorithm's second phase involves the implementation of a controlled rotation that is conditioned on $\langle\lambda_j\rangle$. To do this, the system's third ancilla register is inserted in state $\langle 0\rangle$. The controlled Pauli-Y rotation is then executed, producing a normalized state with the form

$$\sqrt{1-\frac{C^2}{\tilde{\lambda}_j^2}}\left|\tilde{\lambda}_j\right\rangle|u_j\rangle|0\rangle+\frac{C}{\tilde{\lambda}_j}\left|\tilde{\lambda}_j\right\rangle|u_j\rangle|1\rangle \qquad \textit{Ean. 6}$$

where C is the normalization constant. This can be done by applying the operator

$$e^{-i\theta\sigma_y}=\begin{pmatrix}\cos\theta & -\sin\theta\\ \sin\theta & \cos\theta\end{pmatrix} \qquad \textit{Eqn. 7}$$

where $\theta = tan - 1(\frac{C}{\tilde{\lambda}})$.

$A = \sum_j \lambda_j |u_j\rangle\langle u_j|$ therefore $A^{-1} = \sum_j (1/\lambda_j)|u_j\rangle\langle u_j|$ Assumed to be provided with a quantum state $|b = \sum_i b_i|i\rangle$. The Eigen Basis can be used to express this situation, u_j of operator A such that $|b\rangle = \sum_i \beta_i |u_j\rangle$Applying the previously mentioned process to this superposition state yields the state

$$\sum_{j=1}^{N}\beta_j\left|\tilde{\lambda}_j\right\rangle|u_j\rangle\left(\sqrt{1-\frac{C^2}{\tilde{\lambda}_j^2}}|0\rangle+\frac{C}{\tilde{\lambda}_j}|1\rangle\right) \qquad \textit{Eqn. 8}$$

Gives us

$$|0\rangle\otimes\sum_{j=1}^{N}\beta_j|u_j\rangle\left(\sqrt{1-\frac{C^2}{\tilde{\lambda}_j^2}}|0\rangle+\frac{C}{\tilde{\lambda}_j}|1\rangle\right) \qquad \textit{Eqn. 9}$$

Therefore, by measuring the third register and post-selecting on the result "1," modulo the constant normalization factor C, a quantum state near $|x\rangle = A^{-1}|b\rangle$ may be built in the second register. Here, amplitude amplification may be employed to increase the success probability rather than just measuring and postselecting.Notably, Tang's 2018 thesis, A quantum-inspired classical algorithm for recommendation systems [48], basically proved that it is possible to dequantize and solve several linear algebra problems using equally quick classical algorithms. Prior to this, it was thought that there was no time complexity equivalent to HHL for these types of problems. Furthermore, Tang's algorithm's only restriction is the ability to access samples and queries, which is significantly more acceptable than HHL's requirement for rapid state construction. This does not, however, mean that the HHL is no longer relevant. It is important to remember that Tang's algorithm is designed specifically for low-dimensional matrices, whereas the original HHL was intended for sparse matrices. As of right now, the most useful algorithms in literature for low-dimensional problems are those based on quantum machine learning. However, creating arbitrary quantum evolutions for state preparation is still quite challenging [49-53].

4.2 Quantum Support Vector Machine

One of the most crucial techniques in machine learning nowadays is data categorization. It may be applied to recognize, classify, and analyze fresh data. Computer vision issues [54], medical imaging [55, 56], drug discovery [57], handwriting recognition [58], geostatistics [59], and many more domains have made use of these machine learning classification techniques. Classification tools are equipped with machines that can recognize data and, as a result, know how to respond to specific data. Support Vector Machines (SVM) are one of the most widely used techniques for data categorization in machine learning. Because it can divide data into two categories given an input set of training data and a hyperplane drawn between the two categories, the Support Vector Machine (SVM) is very helpful in this regard. It has been possible to theoretically [60]and experimentally [61] recreate quantum SVM machines. These computers tackle our challenges by using qubits rather than conventional bits. Numerous methods for quantum SVM [62–64] and quantum-inspired SVM [65, 66] have been created. The training data for a support vector machine will be provided as $\vec{\{x_1, y_1\}}, \vec{\{x_2, y_2\}} \dots \vec{\{x_n, y_n\}}$ according to the form $\left\{\vec{\{x_i, y_i\}}: \vec{x_i} \in \mathbb{R}^N, y_i = \pm 1\right\}_{i=1,2..n}$ where $\vec{x_i}$ indicated the location of the point int the space R^N and y_i assigns a class value of +1 or –1 to the data, corresponding to its assigned class. If the data can be divided linearly, one of the easiest methods to split it is to use any plane that can satisfy the equation.

$$\vec{w}.\vec{x}_i - b = 0 \qquad \textit{Eqn. 10}$$

The offset from the origin is denoted by $\frac{b}{|\vec{w}|}$, and the vector $\vec{w}$ is the hyperplane's normal. It is occasionally feasible for several planes to meet this equation. In this case, we attempt to design a hard margin support vector machine (SVM) by building two parallel hyperplanes with a maximum distance of 2|w| between them. These hyperplanes are constructed in such a way that for y_i =1, $\vec{w}.x_i - b \geq 1$ and for y_i=-1, $\vec{w}.x_i - b \leq 1$. This may be expressed as $y_i(\vec{w}.x_i - b) \geq 1$. Figure 2 shows how well this hyperplane distinguishes between the two different kinds of data points.

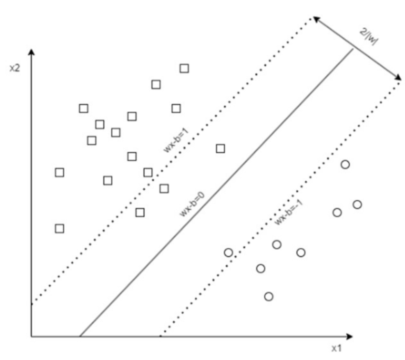

Figure 2 Maximum margin Hyperplane for an SVM

Although the previously described approach may frequently divide data into two groups, the method is typically inapplicable when dealing with nonlinear data. Such issues are typically solved by applying the kernel approach, which raises the problem to a higher dimension and makes it simple to solve using a hyperplane. Utilizing Lagrangian multipliers ($\alpha = (\alpha_1, \alpha_2 ... \alpha_n)$) and solving the dual formulation for optimization are required to do this. Therefore, we may apply the formula to obtain our answer

$$\max\left(\sum_{i=1}^{n}\alpha_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i y_i \alpha_j y_j K_{ij}\right) \quad \textit{Eqn. 11}$$

The decision function of the hyperplane becomes as follows: where K_{ij} is the kernel matrix and the dot product of space is provided as $K_{ij} = \vec{x}_i.\vec{x}_j$, subject to the constraints $\sum_{i=1}^{n} = \alpha_i y_i \geq 0$.

$$f(x) = \mathrm{sgn}\left(\sum_{i=1}^{n}\alpha_i y_i K(x_i, x) + b\right) \quad \textit{Eqn. 12}$$

Also, we can depict it as $w = \sum_{i=0}^{n}\alpha_i y_i . x_i$. Therefore, an SVM may be used to handle nonlinear problems as well. However, occasionally the sheer number of dimensions required to solve a problem unintentionally becomes extraordinarily large. This results in what is known as the Curse of Dimensionality, which is characterized by over-fitting and increased complexity because of a sparse matrix that defines the position of data points getting smaller. The quantum SVMs become significant at this point. Our data is sorted very well using quantum SVM algorithms, which give quantum computers an exponential speedup over conventional computers for solving higher-dimensional issues. However, for a quantum computer to solve it, we must transform our solution into a format that a quantum algorithm can easily understand. One of the most important things to accomplish at this stage is to give the algorithm a certain range of misclassification to prevent over-fitting, and to establish a variable ξ_i that we can use to quantify the misclassification where $\xi_i \geq 0$. Now that we have the following optimization issue written out,

$$\min\left(\frac{1}{2}||w|| + C\sum_{i=1}^{n}\xi_i\right) \quad \textit{Eqn. 13}$$

The cost parameter is represented by C. Consider $C = \gamma/2$ as well, where γ is a cost parameter. After these numbers are determined, we may express our equation as follows: $\vec{w}.\vec{x}_i - b = 1 - \xi_i$. Using the limitations we have been provided with we may find the saddle points of the preceding equation to obtain the equation.

We now need to convert our algorithm into a quantum one to solve our classical algorithm on our quantum computer.

$$F\begin{pmatrix} b \\ \vec{\alpha} \end{pmatrix} = \begin{pmatrix} 0 & 1^T \\ 1 & K + \gamma^{-1}I \end{pmatrix}\begin{pmatrix} b \\ \vec{\alpha} \end{pmatrix} = \begin{pmatrix} 0 \\ y_i \end{pmatrix} \quad \textit{Eqn. 15}$$

First, we will transform our training instances to x_i quantum states in order to do this. We will now translate our matrix $F = J + K_\gamma$, where,

$$J = \begin{pmatrix} 0 & 1^T \\ 1 & 0 \end{pmatrix} \qquad K_\gamma = \begin{pmatrix} 0 & 0 \\ 0 & K + \gamma^{-1}I \end{pmatrix} \quad \textit{Eqn. 14}$$

We will now normalize F as $\hat{F} = \frac{F}{tr(F)} = \frac{F}{tr(K_\gamma)}$ and now we obtain our equation using the Baker-Campbell-Hausdorff formula as

$$e^{-i\hat{F}\Delta t} = e^{\frac{-iJ\Delta t}{K_\gamma}}.e^{\frac{-i\gamma^{-1}I\Delta t}{K_\gamma}}.e^{\frac{-iK_\gamma\Delta t}{K_\gamma}} \quad \textit{Eqn. 16}$$

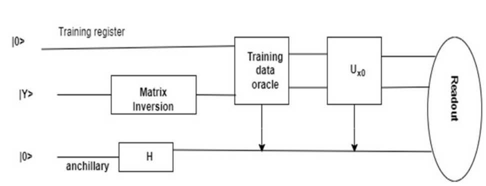

Figure 5 Circuit of quantum SVM

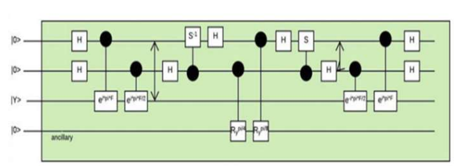

Figure 4 Matrix inversion

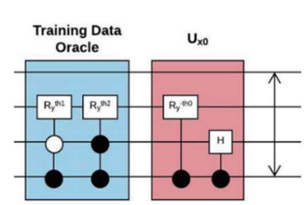

Figure 3 training oracle data and U_{x0}

This reduces our problem to a form where the required values of b and α may be found by finding the eigenvalues and eigen basis of our equation. We can now locate the hyperplane as a result. The exponential improvement in execution speed is one of the primary benefits of employing the quantum SVM [67]. Although this approach is limited to dense training vectors, other techniques have demonstrated efficacy with sparser training vectors. A circuit diagram of this can also be made (Fig. 3). The hyperplane's parameters are obtained in this circuit by the application of matrix inversion. The training data is then entered. Once this is finished, we input the data x_0 to determine which categorization our data is in. You can draw these as (Fig. 4 and 5). Where th_1 and th_2 are the training data, th0 is the data of the location of U_{x0}, and F is the (M+ 1) × (M + 1) matrix containing the component of the Kernel K. Thus, it is evident that one of the best techniques for data classification is the use of quantum SVMs. When it comes to doing data classification, these equations are faster than any other technique. In most systems, they are also easily implementable. These methods do have certain drawbacks, though. First, these algorithms frequently greatly overfit data. This may cause every data point to function as a support vector. This is undesirable and may cause problems when working with big data sets. Additionally, it might make the hyperplane extremely stiff, giving little room for mistakes. We would have to increase a soft SVM's range. Second, these systems work well with linear and polynomial kernels, but they may cause issues with other kernels. Even though most of these systems are linear or polynomial non-symmetrical kernels don't usually create issues, however they may in the future. They also represent a big problem in the future. The general kernel issue will be very important to solve.

4.3 Quantum Classifier

A quantum classifier is an algorithm for quantum computing that classifies new data into the appropriate classes based on the quantum states of the data that already exists. The background research on quantum classifiers and their implementation on quantum computers are covered in the part that follows.

4.3.1 Current Work on quantum classifier

Microsoft [68] recently published a study that proposed a quantum framework for variational-based supervised learning (Fig. 6). The quantum processing unit, or QPU, carries out additional tasks using a model circuit U_θ, a state preparation circuit S_x that converts input x into amplitudes, and single-qubit measurement. Such a QPU measures the amplitudes created by the state preparation circuit and acted upon by the model circuit to do inference using the model, or mathematically determine f (x, θ) = y. From the 0 or 1 that these measurements provide, a binary forecast may be made. A variational technique may be used to train the learnable classification circuit parameters θ. The model circuit translates an encoded feature vector represented by an n-dimensional ket vector ψ_x to another ket ψ.= $U_\theta\psi(x)$ by U_θ, where U_θ is parameterized by θ and is inevitably unitary (Fig. 7). Code blocks B1 and B3 of the circuit above have controls r = 1 and r = 3, respectively. The underlying quantum hardware requires the decomposition of the 17 trainable single-qubit gates G = G (α, β, γ) and the 16 trainable controlled single-qubit gates C(G) into an elementary constant gate set in order to utilize them. When optimization techniques are used to reduce every controlled gate to a single parameter, In the model circuit, we obtain 3 × 33 + 1 = 100 learnable parameters, which may then categorize between inputs corresponding to $2^8 = 256$ dimensions. This model's flexibility makes it far more compact than traditional feed-forward neural networks. In their study, Farhi and Neven [52]described a quantum neural network (QNN) that can be taught using supervised learning techniques and may represent labelled classical or labelled quantum data.

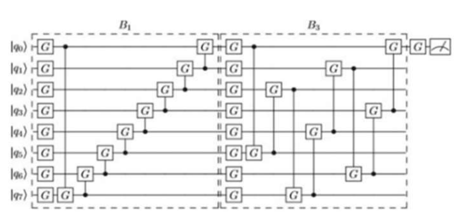

Figure 6 Idea of circuit centric quantum classifier [65]

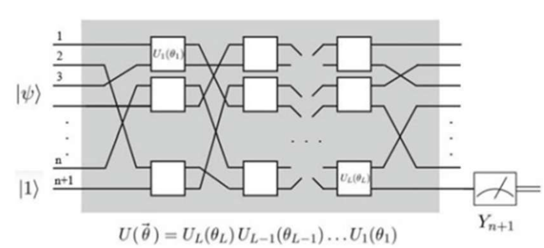

Figure 7 generic model circuit architecture for qubits[65]

For example, a data collection may have the strings $z = z_1z_2...z_n$, where each z_i denotes a bit that can have a value of +1 or −1, and a binary label l(z) that can also have a value of +1 or −1. Furthermore, a quantum processor operating on n + 1 qubits is known to exist (ignoring the potential that ancilla qubits may be needed). The last qubit will serve as a readout. On the input states that are provided, this processor performs unitary transformations; the unitaries that we have come from a toolbox of unitaries, maybe chosen based on experimental considerations [69] The input state $\{\psi, 1\}$ is now prepared, and depending on parameters Θ_i, it is then transformed using a series of qubit unitary transformations $U_i(\Theta_i)$.

The desired label for ψ is produced by measuring Y_{n+1} on the readout qubit, and they are automatically modified during the process. The use of hierarchy-structured quantum circuits for binary classification of classical data encoded in a quantum state is covered in a study by Grant et al. [70]. The authors effectively apply their more expressive circuit designs to the challenge of classifying highly entangled quantum states. These circuits resemble trees and are parameterized by a very basic set of gates that can be handled by quantum computers that are now on the market. A tree tensor network (TTN) is the first of these [71]. We also consider a more intricate circuit design known as the multiscale entanglement renormalization ansatz, or MERA [72]. MERAs are distinct from TTNs in that they make better use of extra unitaries to capture quantum correlations. The literature has proposed TTN and MERA circuits in both one-dimensional (1D) and two-dimensional (2D) variants [73, 74] (Fig. 9).

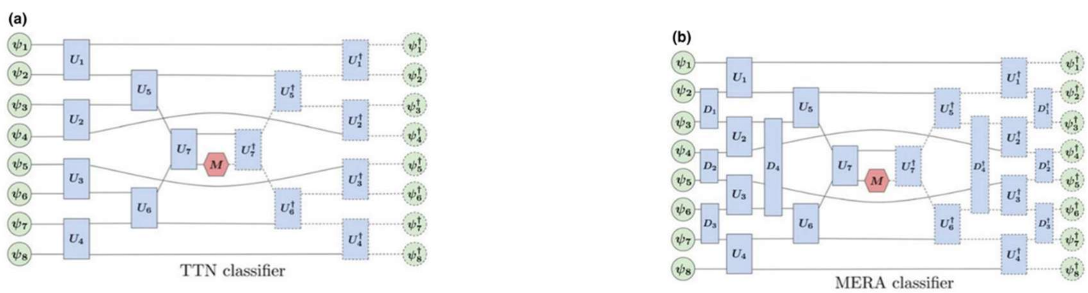

Figure 8 TTN and MERA classifiers for eight qubits. a TNN Classifier, b MERA classifier

Prior work on steady state quantum classifier by Turkpençe et al. [75] shows how to take use of the additivity and divisibility characteristics of fully positive (CP) quantum dynamical maps. Additionally, he provides a numerical demonstration of how a steady-state quantum unit operates as a quantum data classifier in response to varying information environments. The transition of pure quantum states into mixed steady states is impacted by dissipative surroundings' effects on the reduced system dynamics [76]. These mixed states lack a quantum signature and are composed of mixes of conventional probability distributions. This model was used to show how a tiny quantum system, when left in contact with certain quantum environments, may be effective in identifying data contained in these environments.

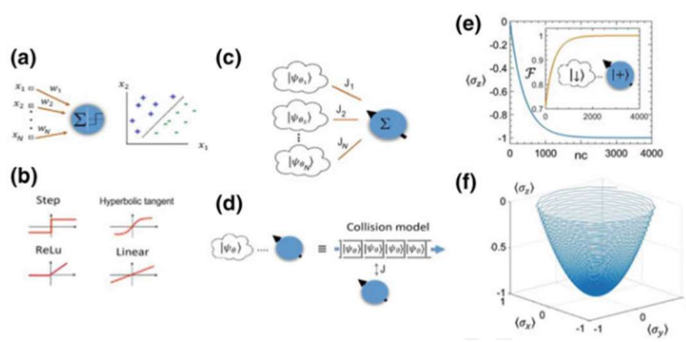

Figure 9 An overview of the suggested model. (a) An N-input classical perceptron. (b) A selection of the perceptron's activation functions. (c) The suggested quantum classifier's architecture. (d) Quantum dynamic system simulation using a collision model. (e) The single spin magnetization's time evolution as a function of collision frequency. (f) The single spin's Bloch ball vector trajectory during the development.

A few frequently used activation functions are shown in Figure 10b. A step function, for example, returns f(y)= −1 otherwise and f (y) = 1 if y =$\sum_i x_i w_i \geq 0$. Following these findings, a correctly operating perceptron is indicated if a line appropriately divides the data instances. The authors establish a connection between the data reservoir and the single spin in the $\rho_\pi = |\downarrow\rangle \langle\downarrow|$ fixed quantum state to benchmark these computations. Next, they apply data is seen in Figure 10d. The authors note that spin's time progression as the spin density matrix gets closer to $(\sigma_z(t)) = -1$, magnetization converges to the fidelity unit with the fix reservoir state monotonically

$$F(t) = \mathrm{Tr}\sqrt{\sqrt{\rho\pi\sigma S(t)}\sqrt{\rho\pi}} = 1 \quad \textit{Eqn. 17}$$

5. Conclusion and Future Work

In the past two decades, quantum computation has advanced substantially in both theoretical foundations and experimental realizations. Numerous algorithms, such as Shor's for factoring and Grover's for database search, have demonstrated the theoretical promise of quantum speedups over classical methods. In parallel, experimental platforms like superconducting qubits, trapped ions, photonic processors, and nuclear magnetic resonance systems have emerged as viable candidates for building quantum computers. In the context of machine learning, quantum computing offers compelling possibilities for accelerating complex computations, managing large datasets, and solving optimization problems that are intractable for classical systems. Quantum algorithms like Quantum Support Vector Machines (QSVM), Quantum Neural Networks (QNNs), Quantum Simulators, and the HHL algorithm for solving linear systems of equations highlight the transformative potential of quantum-enhanced machine learning. However, several challenges persist. One of the foremost limitations lies in the input-output bottlenecks: encoding large-scale classical data into quantum states remains resource-intensive, and decoding quantum results into classical interpretations is similarly challenging. Additionally, hardware-related issues such as decoherence, error rates, limited qubit counts, and constraints on circuit depth restrict the practical deployment of quantum algorithms on current devices. The importance of quantum error correction and fault-tolerant computing cannot be overstated. As quantum systems scale up, robust mechanisms for error detection and correction must evolve, integrating improvements in both software and hardware domains [77] Moreover, most of the current research and development is still predominantly driven by the physics community. To fully harness quantum computing for machine learning applications, deeper collaboration across computer science, applied mathematics, and data science communities is essential [78] Moving forward, several avenues hold promise for advancing this field:

- **Hybrid Quantum-Classical Systems:** Development of hybrid frameworks where quantum subroutines are integrated into classical machine learning pipelines. These systems could optimize the use of quantum resources while offloading certain tasks to classical processors.
- **Quantum Data Encoding Techniques:** Improved quantum feature maps and data loading schemes are required to reduce the overhead associated with data encoding and state preparation, making quantum algorithms more scalable and practical [79].
- **Advanced Quantum Algorithms for Nonlinear Models:** Existing quantum machine learning algorithms are primarily adapted for linear models. Developing quantum-native approaches for nonlinear problems, neural networks, and generative models remains a significant research opportunity.
- **Error-Tolerant Machine Learning Models:** Future work should explore algorithms inherently resilient to noise and decoherence. This might involve designing models and loss functions that can tolerate a degree of quantum error without degrading performance.
- **Real-World Application Demonstrations:** Quantum algorithms must eventually be tested on real-world, domain-specific problems, such as drug discovery, financial modeling, logistics optimization, and materials science, moving beyond synthetic benchmarks.
- **Quantum Cloud Infrastructure:** The growth of quantum cloud platforms (IBM Quantum Experience, Azure Quantum, etc.) offers new opportunities for distributed, collaborative research and benchmarking of quantum machine learning algorithms in live environments.
- **Interdisciplinary Collaboration:** Establishing interdisciplinary research initiatives combining expertise from quantum physics, computer science, artificial intelligence, mathematics, and engineering to collectively address the theoretical and practical challenges of quantum-enhanced machine learning.

In conclusion, while significant hurdles remain, the intersection of quantum computing and machine learning presents one of the most exciting frontiers in computational science. Continued advancements in hardware, algorithms, and collaborative research frameworks will be essential to transition these technologies from experimental prototypes to impactful, real-world applications.

References

1. Yan, S. Y. (2018). Cybercryptography: Applicable Cryptography for Cyberspace Security. Germany: Springer International Publishing.
2. Shor, P. W. (1999). Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM review*, *41*(2), 303-332.
3. Grover, L. K. (1996, July). A fast quantum mechanical algorithm for database search. In *Proceedings of the twenty-eighth annual ACM symposium on Theory of computing* (pp. 212-219).
4. Shor, P. W. (1994, November). Algorithms for quantum computation: discrete logarithms and factoring. In *Proceedings 35th annual symposium on foundations of computer science* (pp. 124-134). Ieee.
5. Lloyd, S. (1996). Universal quantum simulators. *Science*, *273*(5278), 1073-1078.
6. Wittek, P. (2014). *Quantum machine learning: what quantum computing means to data mining*. Academic Press.
7. Planck, M. (1901). On the law of distribution of energy in the normal spectrum. *Annalen der physik*, *4*(553), 1.
8. Muhyedeen, B. R. (2008). On Heuristic Viewpoints Concerning the Mass, Energy and Light Concepts in Quantum Physics. *Euro. J. Sci. Res*, *22*(4), 584-601.
9. Bohr, N. (1913). I. On the constitution of atoms and molecules. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, *26*(151), 1-25.
10. Hentschel, Klaus. (2009). Stark Effect. 10.1007/978-3-540-70626-7_209.
11. Weinberger, P.. (2006). Revisiting Louis de Broglie's famous 1924 paper in the Philosophical Magazine. Philosophical Magazine Letters - PHIL MAG LETT. 86. 405-410. 10.1080/09500830600876565.
12. Heisenberg, W. (1984). Quantum Theoretical Kinematics and Mechanics. *Scientific Review*
13. Schrödinger, E. (1926). Quantization as an eigenvalue problem. *Annalen der Physik*, *79*(4), 361-376.
14. Schrödinger, E. (1926). An undulatory theory of the mechanics of atoms and molecules. *Physical review*, *28*(6), 1049.
15. Bell, J. S. (1964). On the einstein podolsky rosen paradox. *Physics Physique Fizika*, *1*(3), 195.
16. Mukta Nivelkar and S. G. Bhirud 2022 *J. Phys.: Conf. Ser.* 2161 012023
17. Hey, T. (1999). Quantum computing: an introduction. *Computing & Control Engineering Journal*, *10*(3), 105-112.
18. Feynman, R. P. (2018). *Feynman lectures on computation*. CRC Press.
19. Feynman, R. (1982). Simulating Physics with Computers International Journal of Theoretical Physics–Springer Science+ Business Media.
20. Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer by Peter Shor in 1997.
21. A Fast Quantum Mechanical Algorithm for Database Search by Lov Grover in 1996.
22. Aïmeur, E., Brassard, G., Gambs, S. (2006). Machine Learning in a Quantum World. In: Lamontagne, L., Marchand, M. (eds) Advances in Artificial Intelligence. Canadian AI 2006. Lecture Notes in Computer Science(), vol 4013. Springer, Berlin, Heidelberg.
23. S. Shahane, S. Shendye, A. Shaikh, Implementation of artificial neural network learning methods on embedded platform. Int. J. Electric. Electron. Comput. Syst. 2, 2347 (2014)
24. Lloyd, S., Mohseni, M., & Rebentrost, P. (2013). Quantum algorithms for supervised and unsupervised machine learning. *arXiv preprint arXiv:1307.0411*.
25. Buhrman, H., Cleve, R., Watrous, J., & De Wolf, R. (2001). Quantum fingerprinting. *Physical Review Letters*, *87*(16), 167902.
26. Wiebe, N., Kapoor, A., & Svore, K. (2014). Quantum algorithms for nearest-neighbor methods for supervised and unsupervised learning. *arXiv preprint arXiv:1401.2142*.

27. Rebentrost, P., Mohseni, M., & Lloyd, S. (2014). Quantum support vector machine for big data classification. *Physical review letters*, *113*(13), 130503.
28. Clark, L. A., Huang, W., Barlow, T. M., & Beige, A. (2015). Hidden quantum markov models and open quantum systems with instantaneous feedback. In *ISCS 2014: Interdisciplinary Symposium on Complex Systems* (pp. 143-151). Springer International Publishing.
29. Chatterjee, R., & Yu, T. (2016). Generalized coherent states, reproducing kernels, and quantum support vector machines. *arXiv preprint arXiv:1612.03713*.
30. Zhao, Z., Fitzsimons, J. K., & Fitzsimons, J. F. (2019). Quantum-assisted Gaussian process regression. *Physical Review A*, *99*(5), 052331.
31. Rigatos, G. G., & Tzafestas, S. G. (2007). Neurodynamics and attractors in quantum associative memories. *Integrated Computer-Aided Engineering*, *14*(3), 225-242.
32. E. C. Behrman and J. E. Steck, "A quantum neural network computes its own relative phase," 2013 IEEE Symposium on Swarm Intelligence (SIS), Singapore, 2013, pp. 119-124, doi: 10.1109/SIS.2013.6615168.
33. Gupta, S., & Zia, R. K. P. (2001). Quantum neural networks. *Journal of Computer and System Sciences*, *63*(3), 355-383.
34. Adachi, S. H., & Henderson, M. P. (2015). Application of quantum annealing to training of deep neural networks. *arXiv preprint arXiv:1510.06356*.
35. Biamonte, J., Wittek, P., Pancotti, N., Rebentrost, P., Wiebe, N., & Lloyd, S. (2017). Quantum machine learning. *Nature*, *549*(7671), 195-202.
36. Schuld, M., Sinayskiy, I., & Petruccione, F. (2014). The quest for a quantum neural network. *Quantum Information Processing*, *13*, 2567-2586.
37. Monras, A., Beige, A., & Wiesner, K. (2010). Hidden quantum Markov models and non-adaptive read-out of many-body states. *arXiv preprint arXiv:1002.2337*.
38. Clark, L. A., Huang, W., Barlow, T. M., & Beige, A. (2015). Hidden quantum markov models and open quantum systems with instantaneous feedback. In *ISCS 2014: Interdisciplinary Symposium on Complex Systems* (pp. 143-151). Springer International Publishing.
39. Srinivasan, S., Gordon, G., & Boots, B. (2018, March). Learning hidden quantum Markov models. In *International Conference on Artificial Intelligence and Statistics* (pp. 1979-1987). PMLR.
40. Buluta, I., & Nori, F. (2009). Quantum simulators. *Science*, *326*(5949), 108-111.
41. Hassija, V., Chamola, V., Goyal, A., Kanhere, S.S. and Guizani, N. (2020), Forthcoming applications of quantum computing: peeking into the future. IET Quantum Commun., 1: 35-41.
42. Biamonte, J., Wittek, P., Pancotti, N., Rebentrost, P., Wiebe, N., & Lloyd, S. (2017). Quantum machine learning. *Nature*, *549*(7671), 195-202.
43. A.M. Childs, R. Kothari, R.D. Somma, SIAM J. Comput. 46, 1920 (2017)
44. B.D. Clader, B.C. Jacobs, C.R. Sprouse, Preconditioned quantum linear system algorithm. Phys. Rev. Lett. 110.25, 250504 (2013)
45. D. Dervovic et al., Quantum Linear Systems Algorithms: A Primer (2018).
46. S. Dutta et al., Demonstration of a Quantum Circuit Design Methodology for Multiple Regression (2018)
47. E. Tang, A quantum-inspired classical algorithm for recommendation systems, in Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing (ACM, 2019)
48. E. Tang, Quantum-inspired classical algorithms for principal component analysis and supervised clustering (2018)
49. A. Gilyén, S. Lloyd, E. Tang, Quantum-inspired low-rank stochastic regression with logarithmic dependence on the dimension (2018).
50. N.-H. Chia, H.-H. Lin, C. Wang C, Quantum-inspired sublinear classical algorithms for solving low-rank linear systems (2018)
51. E. Tang, An overview of quantum-inspired classical sampling (2019)
52. E. Tang, Some settings supporting efficient state preparation (2019).
53. S. Nowozin, C.H. Lampert, Structured learning and prediction in computer vision. Found. Trends Comput. Graph. Vis. 6, 185–365 (2011)
54. M.N. Wernick, Y. Yang, J.G. Brankov et al., Machine learning in medical imaging. IEEE 27, 25–38 (2010)
55. (2010) 37. B.J. Erickson, P. Korfiatis, Z. Akkus, T.L. Kline, Machine learning for medical imaging. Rad. Graph. 37, 505–515 (2017)
56. A. Lavecchia, Machine-learning approaches in drug discovey: methods 5nd applications. Sci. Dir. 20, 318–331 (2015)

57. C. Bahlmann, B. Haasdonk, H. Burkhardt, Online handwriting recognition with support vector machines—a kernel approach, in Proceedings Eighth International Workshop on Frontiers in Handwriting Recognition (2002)
58. L. Chen, C. Ren, L. Li et al., A comparative assessment of geostatistical, machine learning, and hybrid approaches for mapping topsoil organic carbon content. Int. J. Geo-Inf. 8(4), 174 (2019)
59. P. Rebentrost, M. Mohseni, S. Lloyd, Quantum support vector machine for big data classification. Phys. Rev. Lett. 113, 130503 (2014)
60. Z. Li, X. Lui, N. Xu, J. Du, Experimental realization of a quantum support vector machine. Phys. Rev. Lett. 114, 140504 (2015)
61. I. Kerenidis, A. Prakash, D. Szilágyi, Quantum algorithms for second-order cone programming and support vector machines
62. A.K. Bishwas, A. Mani, V. Palade, Big data quantum support vector clustering (2018). arXiv:1804.10905
63. Arodz, T., Saeedi, S. Quantum sparse support vector machines (2019).
64. C. Ding, T. Bao, H. Huang, Quantum-inspired support vector machine (2019)
65. D. Anguita, S. Ridella, F. Rivieccio, R. Zunino, Quantum optimization for training support vector machines. Neural Netw. 16, 763–770 (2003)
66. B.J. Chelliah, S. Shreyasi, A. Pandey, K. Singh, Experimental comparison of quantum and classical support vector machines. IJITEE 8, 208–211 (2019)
67. M. Schuld, A. Bacharov, K. Svore, N. Wiebe (2018)
68. M. Schuld, I. Sinayskiy, F. Petruccione, An introduction to quantum machine learning. Cont. Phys. 56, 172–185 (2015)
69. E. Grant, M. Benedetti, S. Cao, A. Hallam, J. Lockhart, V. Stojevic, A. Green, S. Severini, npj Quantum Inf. 4, 65 (2018)
70. Y.-Y. Shi, Y.-Y., Duan, L.-M., and Vidal,G., Classical simulation of quantum many-body systems with a tree tensor network, Phys. Rev. A 74, 022320 (2006)
71. G. Vidal, Class of quantum many-body states that can be efficiently simulated. Phys. Rev. Lett. 101, 110501 (2008)
72. L. Cincio, J. Dziarmaga, M.M. Rams, Multiscale entanglement renormalization ansatz in two dimensions: quantum ising model. Phys. Rev. Lett. 100, 240603 (2008)
73. G. Evenbly, G. Vidal, Entanglement renormalization in noninteracting fermionic systems. Phys. Rev. B 81, 235102 (2010)
74. D. Turkpençe et al., A Steady state quantum classifier. Phys. Lett. A 383, 1410 (2019)
75. H.P. Breuer, F. Petruccione, The Theory of Open Quantum Systems (Oxford University Press, Oxford, 2007)
76. J. Biamonte, P. Wittek, N. Pancotti, P. Rebentrost, N. Wiebe, and S. Lloyd, "Quantum machine learning," Nature, vol. 549, no. 7671, pp. 195–202, 2017.
77. Preskill, J. (2018). *Quantum computing in the NISQ era and beyond*. Quantum, 2, 79.
78. Schuld, M., Sinayskiy, I., & Petruccione, F. (2015). *An introduction to quantum machine learning*. Contemporary Physics, 56(2), 172-185.
79. Benedetti, M., Lloyd, E., Sack, S., & Fiorentini, M. (2019). *Parameterized quantum circuits as machine learning models*. Quantum Science and Technology, 4(4), 043001.