



## OPEN ACCESS

## EDITED BY

Ting Yu,  
Stevens Institute of Technology, United States

## REVIEWED BY

Siva Subramanian R,  
SRM Institute of Science and Technology  
(Deemed to be University) Research  
Kattankulathur, India  
Yevhen Vasiliiu,  
State University of Intelligent Technologies and  
Telecommunications, Ukraine

## \*CORRESPONDENCE

V. Raseena,  
✉ raseenav@siasindia.org

RECEIVED 12 October 2025

REVISED 24 November 2025

ACCEPTED 08 December 2025

PUBLISHED 29 December 2025

## CITATION

Raseena V (2025) Quantum computing:  
foundations, algorithms, and  
emerging applications.  
*Front. Quantum Sci. Technol.* 4:1723319.  
doi: 10.3389/frqst.2025.1723319

## COPYRIGHT

© 2025 Raseena. This is an open-access article distributed under the terms of the [Creative Commons Attribution License \(CC BY\)](#). The use, distribution or reproduction in other forums is permitted, provided the original author(s) and the copyright owner(s) are credited and that the original publication in this journal is cited, in accordance with accepted academic practice. No use, distribution or reproduction is permitted which does not comply with these terms.

# Quantum computing: foundations, algorithms, and emerging applications

V. Raseena\*

SAFI Institute of Advanced Study (Autonomous), Malappuram, India

Quantum computing is an emerging paradigm that leverages the principles of quantum mechanics to solve computational problems beyond the reach of classical computers. This article provides an overview of the fundamental concepts of qubits, the distinctive features of quantum mechanics such as superposition and entanglement, and the challenges of building scalable, fault-tolerant systems. It surveys key quantum algorithms and their potential applications in fields including cryptography, optimization, finance, chemistry, and machine learning. Additionally, it highlights the importance of verification frameworks for ensuring the reliability of quantum programs. A literature review of significant contributions is presented, drawing insights from recent surveys on quantum algorithms, qubit technologies, and software verification approaches. The article concludes by discussing ongoing challenges, such as error correction overhead, hardware scalability, and verification complexity, and suggests directions for future research.

## KEYWORDS

quantum computing, qubits, quantum algorithms, quantum verification, error correction, quantum applications

## 1 Introduction

Imagine a machine that does not compute by flipping a long string of zeros and ones, but by coaxing tiny quantum objects into behaving like complex waves of possibility. That's the intuitive leap behind quantum computing: instead of bits that are definitely 0 or 1, quantum computers use qubits that can exist in superpositions of states, become entangled so their fates are linked across space, and exploit interference to amplify correct answers while canceling wrong ones. These phenomena: superposition, entanglement, and interference are the conceptual tools that let quantum algorithms explore many possible solutions at once in ways classical algorithms cannot.

Because of those properties, quantum machines have the potential to transform domains where classical approaches struggle. Simulating complex molecules for chemistry and materials science, tackling hard optimization problems in logistics and finance, and accelerating certain kinds of machine-learning and search tasks. Researchers have already demonstrated early milestones where quantum processors performed narrowly defined tasks far faster than classical machines, milestones sometimes called quantum supremacy or quantum advantage. These show the field is progressing from theory toward demonstrable speedups (Benioff, 1980; Feynman, 1982; Deutsch, 1985; Shor, 1995; Grover, 1996).

The current era, often referred to as the NISQ (Noisy Intermediate-Scale Quantum) era, is characterized by machines with tens to hundreds of qubits that are inherently noisy and

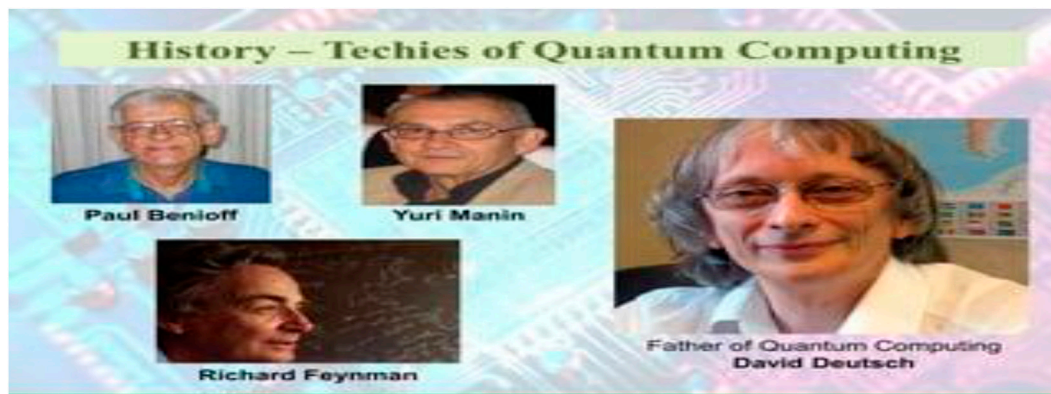


FIGURE 1  
Quantum computing, Slideshare presentation (2021) (See Ref [Quantum Computing, 2021](#)).

prone to errors. Fully fault-tolerant, error-corrected quantum computing remains an engineering challenge, but progress is rapid. Large technology companies and startups alike continue to publish new processors, algorithms, and roadmaps. Quantum computing is no longer just a theoretical curiosity, it is an active, multidisciplinary race among physicists, engineers, and computer scientists to turn exotic quantum effects into practical advantage (Preskill, 2018).

## 2 Literature review

The literature on quantum computing is broad and rapidly evolving, spanning foundational theory, algorithms, hardware, applications, and software/verification tooling. Below we summarize key strands of work and representative references that capture both historical foundations and modern survey-level syntheses.

### 2.1 Foundational theories and early contributions

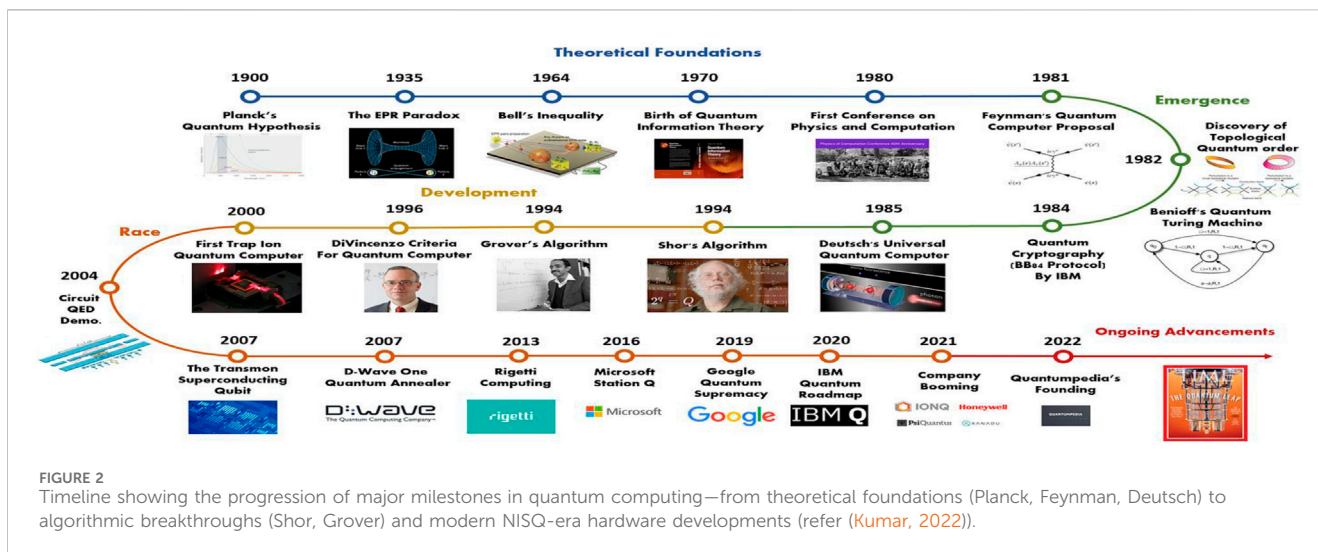
The theoretical foundations of quantum computation were laid in the late 20th century. Yuri Manin was among the first to suggest that computation could be performed according to the principles of quantum mechanics, introducing the concept of quantum automata in his 1980 work *Computable and Uncomputable* (Benioff, 1980). Shortly afterward, Paul Benioff proposed quantum-mechanical models of Turing machines, showing how computation could be embedded in quantum physics (Feynman, 1982). Richard Feynman famously argued that quantum systems are hard to simulate efficiently on classical computers and proposed the idea of using quantum devices to simulate nature, a key motivation for quantum computing (Deutsch, 1985). David Deutsch formalized the notion of a universal quantum computer and articulated the Church–Turing principle in the quantum setting, establishing that quantum mechanics could define a new model of computation (Shor, 1995). These foundational works set the conceptual stage for later algorithms and hardware efforts (Grover, 1996; Brassard et al., 2000). The historical development

and major milestones of quantum computing are illustrated in Figures 1, 2.

### 2.2 Surveys of quantum algorithms and applications

Recent survey work synthesizes how quantum algorithms map onto real-world application areas—such as chemistry, optimization, cryptography, machine learning, and finance—carefully weighing theoretical speedups against practical resource costs and engineering constraints. Dalzell (2023) provide a useful, application-oriented perspective that emphasizes subtle caveats regarding when quantum advantage actually materializes and the need for end-to-end complexity considerations (Dalzell, 2023). Contemporary literature stresses that theoretical asymptotics (e.g., Shor’s exponential speedup) must be evaluated alongside requirements for fault tolerance, qubit counts, and realistic gate/noise budgets (Harrow et al., 2009; Preskill, 2018; Arute et al., 2019).

Beyond these, Huynh et al. (2023) explore *quantum-inspired machine-learning* approaches that bridge classical and quantum paradigms, offering hybrid algorithms deployable on today’s near-term hardware (Huynh et al., 2023). Grigoryan et al. (2025) provides a comprehensive review of quantum-computing models—including gate-based, adiabatic, and measurement-based approaches—analyzing their algorithmic implications and domain-specific applications (Grigoryan et al., 2025). Industry-oriented analyses published in *EPJ Quantum Technology* (2021) emphasize how algorithmic progress interacts with hardware engineering, highlighting persistent gaps between theoretical quantum advantage and practical scalability (EPJ Quantum Technology, 2021). Additionally, the *Quantum Algorithm Zoo* (2025) serves as an evolving catalogue of hundreds of quantum algorithms, classified by domain and computational model, offering researchers a living reference for tracking progress across the field (Quantum Algorithm Zoo, 2025). Together, these surveys illustrate that while theoretical speedups remain intellectually compelling, their translation into practical advantage depends critically on hardware maturity, hybrid algorithm design, and integrated benchmarking frameworks.



## 2.3 Qubit technologies and hardware development

Work on qubit platforms surveys the trade-offs among leading physical implementations: superconducting circuits (offering excellent gate speed and strong industrial support), trapped ions (high fidelity and long coherence but slower gates), photonic systems (room-temperature operation and flexible connectivity), neutral atoms, nitrogen-vacancy (NV) centers, and semiconductor quantum dots. Comprehensive reviews compare coherence times, gate fidelities, and error mechanisms across these platforms (Chae et al., 2024). These hardware surveys are crucial for determining which algorithmic proposals are practical for near-term devices and which depend on long-term fault-tolerant architectures (Peruzzo et al., 2014).

The IBM Quantum Roadmap (2024–2025) details the engineering milestones toward modular and error-corrected systems, highlighting chip-to-chip connectivity, cryogenic control, and scalable architectures (IBM Quantum, 2025; Quantum, 2025). The IBM Osprey processor (2023), a 433-qubit superconducting chip, demonstrated significant progress in coherence control and fabrication uniformity (Preskill, 2018; IBM Quantum, 2023). Broader theoretical and experimental discussions continue to frame such systems within the Noisy Intermediate-Scale Quantum (NISQ) paradigm (Preskill, 2018).

Beyond superconducting and ion-trap systems, hybrid hardware directions are emerging that integrate photonic, atomic, and NV-center qubits for specialized applications such as quantum chemistry and clinical computation (Lee et al., 2024; Sharma et al., 2025; Monroe, 2024; Goswami and Patel, 2025). Educational and technical resources describe two-qubit-gate physics and multi-qubit coupling strategies that are crucial for scalable architectures (Quantum, 2024). Cross-platform benchmarking efforts emphasize that progress in algorithmic performance depends on unified error metrics, calibration protocols, and realistic gate budgets (Gill et al., 2020; Grigoryan et al., 2025; Zhao and Kumar, 2024). Collectively, these analyses underscore that while hardware diversity remains a strength, convergence toward modular, fault-tolerant design is

the key enabler for sustainable quantum advantage in the coming decade.

## 2.4 Verification, software stacks, and reliability

As quantum algorithms and hardware continue to mature, the verification of quantum programs and toolchain correctness has become a rapidly growing research area. Recent surveys summarize formal verification techniques specifically adapted to quantum settings, including model checking, deductive verification using proof systems, compiler-level equivalence checking, and error-correction verification frameworks (Lewis et al., 2021; Qafny, 2025; CoqQ, 2025; CertiQ, 2025; Veri-QEC, 2025). These approaches have been implemented in verification tools such as QPMC, PRISM, STORM, CoqQ, and Veri-QEC, each providing distinct trade-offs in scalability and rigor. Circuit equivalence checking is also supported by dedicated tools such as the Quantum Equivalence Checker (QEC, 2025).

The literature highlights that ensuring quantum software correctness is significantly more challenging than in classical systems because of probabilistic outputs, exponentially large state spaces, entanglement dependencies, and complex noise models (Bauer-Marquart et al., 2022; Gill et al., 2020). To address these difficulties, researchers have introduced symbolic-execution-based and SMT-solver-backed verifiers that automate the discovery of functional and security-related defects. Deductive frameworks and theorem-prover embeddings (e.g., Qafny, QHLProver, and CoqQ) are being developed to enhance formal reasoning about quantum programs, allowing stepwise verification of program semantics and circuit transformations (Qafny, 2025; CoqQ, 2025; QHLProver, 2025).

Recent work also explores the integration of verification techniques into quantum software stacks, ensuring correctness across compilation, optimization, and hardware execution layers (Shi, 2019; Grigoryan et al., 2025; Zhao and Kumar, 2024). Frameworks such as CertiQ and symQV extend traditional

compiler verification into the quantum domain, validating equivalence between logical circuits and their optimized or transpiled forms (Shi, 2019; Bauer-Marquart et al., 2022). Collectively, these studies indicate that formal verification will be a foundational component of future quantum software engineering, bridging the gap between abstract algorithm design and reliable hardware execution.

## 2.5 Application-focused and gap analyses

Domain-specific surveys and studies across finance, chemistry, logistics, and machine learning emphasize persistent gaps between theoretical quantum advantage and experimental feasibility. While algorithmic proposals demonstrate promising asymptotic speedups, end-to-end resource analyses are frequently incomplete, and assumptions about idealized, error-free hardware dominate much of the literature (Herman, 2022; Schuld, 2021). Verification and benchmarking remain at an early stage, with limited experimental validation and inconsistent reporting of quantum resources (Dalzell, 2023; Grigoryan et al., 2025; Zhao and Kumar, 2024).

Recent reviews have highlighted that realistic quantum advantage demands hardware–software co-design, integrating insights from algorithm development, quantum control engineering, and compiler optimization (Lee et al., 2024; EPJ Quantum Technology, 2021; Quantum, 2025; Andersen et al., 2025). Studies in finance and logistics note that problem encodings and quantum data-loading overheads often offset theoretical speedups, calling for transparent resource estimation frameworks (Herman, 2022; Vidal et al., 2025). Similarly, in quantum chemistry and materials science, Grigoryan et al. (2025) and related works underscore the necessity of aligning algorithmic complexity with hardware noise and decoherence limits (Grigoryan et al., 2025; Zhang et al., 2024).

Emerging meta-analyses propose standardized benchmarking and reproducibility protocols for quantum algorithms—such as the QBench and QPack initiatives—which aim to quantify algorithmic efficiency relative to hardware constraints (Carter and Gheorghiu, 2024). Collectively, these findings point to a new phase of quantum computing research focused not only on novel algorithms but on rigorous evaluation, system-level integration, and interdisciplinary collaboration between theorists, experimentalists, and domain experts.

## 2.6 How to read these surveys and next steps for researchers

For newcomers, understanding the evolution of quantum computing requires a layered approach. Foundational papers introduce the conceptual basis: Benioff's quantum Turing model, Feynman's simulation arguments, and Deutsch's universal quantum computer formalism remain essential for grasping the motivation and theoretical framework (Benioff, 1980; Feynman, 1982; Deutsch, 1985). Modern surveys such as Dalzell et al. provide an application-oriented overview, highlighting algorithmic opportunities and end-to-end

complexity analysis across multiple domains (Dalzell, 2023). Hardware reviews help map algorithms to feasible physical platforms, connecting computational models with real-world architectures (Chae et al., 2024; IBM Quantum, 2025; Quantum, 2025). Verification and reliability studies then bridge theory and engineering, outlining the constraints of quantum software stacks and emerging formal tools (Lewis et al., 2021; Qafny, 2025; CoqQ, 2025; Veri-QEC, 2025; Zhou et al., 2024).

For advancing research, recent literature identifies several high-impact directions:

- End-to-end cost modeling that integrates algorithmic, architectural, and physical resource layers (Preskill, 2018; Dalzell, 2023; Veri-QEC, 2025; Andersen et al., 2025);
- Scalable verification frameworks capable of handling large circuits and realistic noise (Lewis et al., 2021; CertiQ, 2025; Zhou et al., 2024; Morales et al., 2025);
- Benchmarking in the NISQ era, with standardized performance and reproducibility protocols (Preskill, 2018; Dalzell, 2023; Zhao and Kumar, 2024; Carter and Gheorghiu, 2024);
- Hardware–software co-design, promoting collaboration between algorithm designers, compilers, and experimentalists (Grigoryan et al., 2025; Quantum, 2025; Andersen et al., 2025; Patel and Zurek, 2025).

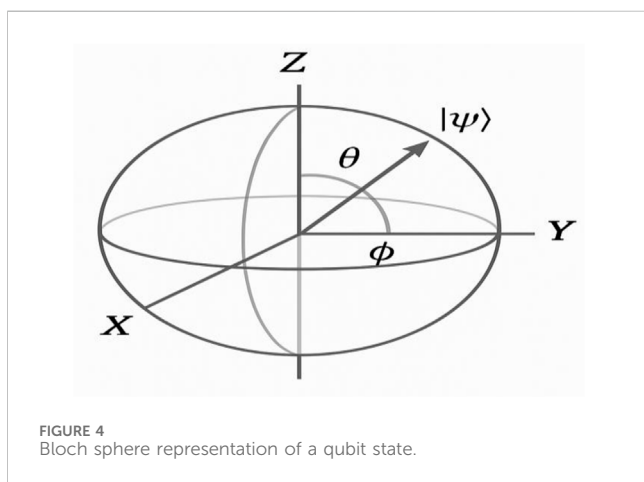
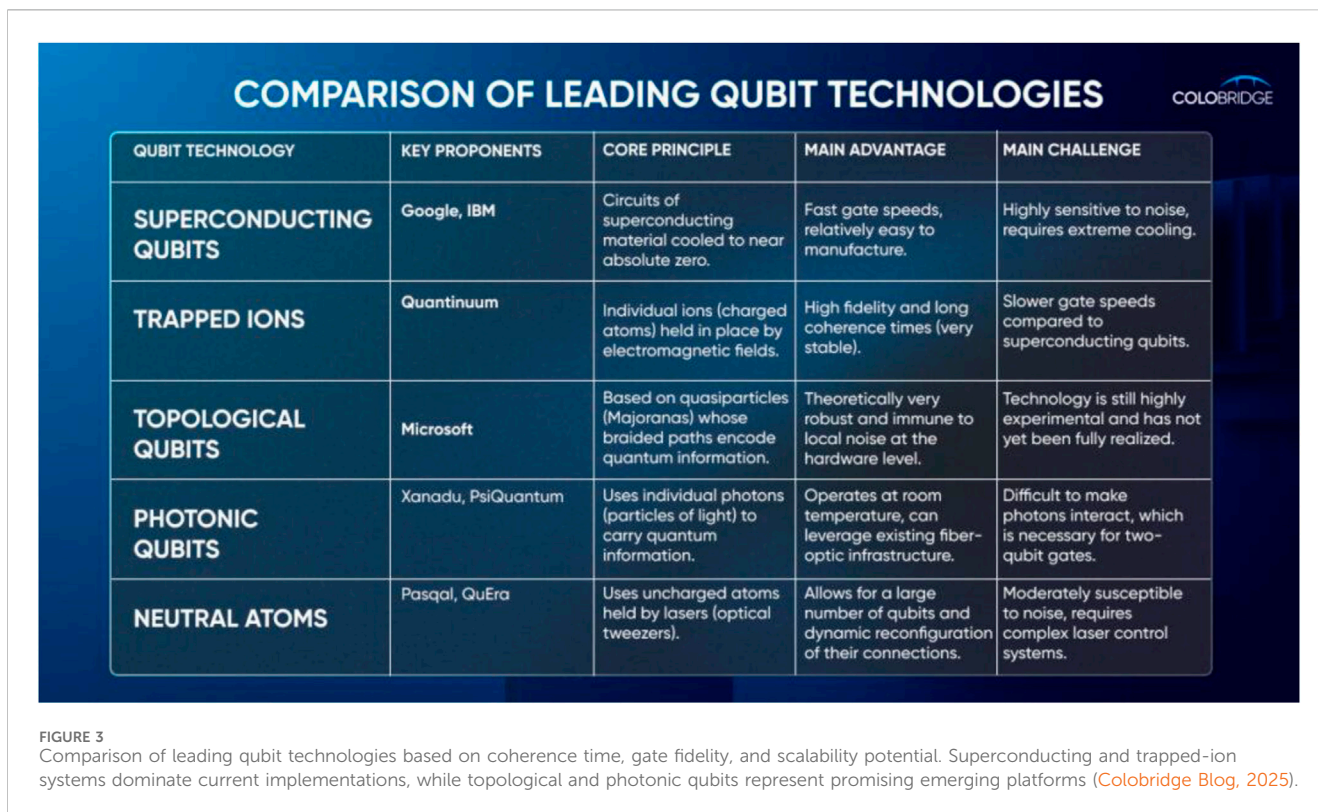
Recent meta-analyses emphasize that bridging theoretical elegance with practical implementation is the next frontier for quantum computing research (Khabiboulline et al., 2025; Reynolds et al., 2025). By sequentially engaging with historical foundations, algorithmic surveys, hardware overviews, and verification frameworks, researchers can develop an integrated understanding of the field and identify tractable research challenges aligned with emerging industrial and scientific priorities.

# 3 Fundamentals of quantum computing

## 3.1 Qubits

In classical computing, information is stored in bits that take values 0 or 1. In quantum computing, the basic unit of information is the quantum bit or qubit, which can exist in a superposition of  $|0\rangle$  and  $|1\rangle$  states. Mathematically, a qubit's state is expressed as  $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$ , where  $\alpha$  and  $\beta$  are complex amplitudes satisfying  $|\alpha|^2 + |\beta|^2 = 1$  (Quantum, 2024). The power of qubits lies in their ability to encode exponentially more information as the number of qubits increases.

Different technologies realize qubits, including superconducting circuits, trapped ions, photonic qubits, topological qubits, and neutral atoms (Chae et al., 2024; Chohan, 2024; AbuGhanem, 2025; Zhu and Browne, 2025; Kim et al., 2025). Comparative studies highlight the trade-offs among these platforms—superconducting qubits offer rapid gate operations and mature fabrication pipelines, while trapped-ion and photonic qubits exhibit superior coherence and connectivity (Chen et al., 2024). Reviews further examine emerging materials, hybrid



processors, and topological error protection as critical frontiers in scalable quantum computing (AbuGhanem, 2025; Zhu and Browne, 2025; Kim et al., 2025). A comparison of leading qubit technologies and their key characteristics is shown in Figure 3.

### 3.2 Superposition

Superposition allows qubits to exist in linear combinations of basis states until measurement collapses them. Unlike classical probability, superposition enables interference—constructive and destructive—allowing quantum algorithms to amplify correct results (Chae et al., 2024). For example, three qubits can simultaneously represent eight states that can be manipulated together by quantum

operations (IBM Quantum, 2025). The geometry of superposition on the Bloch sphere is shown in Figure 4, where a qubit's state  $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$  is represented by polar and azimuthal angles  $\theta$  and  $\phi$  on the sphere's surface (Quantum, 2024).

### 3.3 Entanglement

Entanglement is a uniquely quantum phenomenon in which two or more qubits become correlated such that the state of one determines the state of the others, regardless of spatial separation. This property underpins many quantum algorithms and communication protocols. A well-known example is the *Bell state*,  $(|00\rangle + |11\rangle)/\sqrt{2}$  (QPMC). Without entanglement, many quantum algorithms would lose their exponential advantage (Chohan, 2024; Zhu and Browne, 2025).

### 3.4 Differences from classical computing and challenges

The difference between classical and quantum computing can be visualized using a library analogy: a classical computer checks each shelf sequentially, whereas a quantum computer exploits superposition to explore many shelves simultaneously. Despite this potential, quantum computers face several fundamental challenges:

- Decoherence, or loss of quantum information due to environmental noise;
- High gate error rates, requiring error-correction protocols;
- Scalability issues, as maintaining coherence across many qubits remains difficult;

TABLE 1 Comparison between quantum and classical computing across key operational, algorithmic, and technological aspects.

Aspects	Quantum computing	Classical computing
Basic operation	Processes multiple solutions simultaneously (quantum parallelism)	Processes tasks sequentially
Speed potential	Offers exponential speedups for specific problems like factorization and optimization	Limited by sequential processing
Algorithm advantage	Can leverage quantum algorithms like Grover's and Shor's	Relies on classical algorithms
Speed limitations	Hindered by challenges such as decoherence and error correction	Performs reliably across a wide range of tasks
Current practical speed	Not universally faster due to implementation challenges	Consistently reliable and predictable
Technological development	Advancing rapidly with improvements in qubit coherence	Continues to advance with Moore's law
Applications	Specialized tasks in cryptography, optimization, and simulation	The broad range of applications including everyday computing
Future potential	Holds promise for revolutionizing complex calculations	Likely to remain foundational with continued advancements

TABLE 2 Chronological summary of major theoretical and technological milestones in quantum computing, illustrating the field's evolution from foundational quantum physics to modern multi-qubit architectures.

Year	Milestone/contributor	Key development	Significance
1900	Max Planck	Quantum hypothesis introduced	Foundation of quantum theory
1927	Werner Heisenberg	Uncertainty principle	Defined quantum behavior limits
1935	Albert Einstein, Boris Podolsky, Nathan Rosen	EPR paradox	Introduced concept later known as quantum entanglement
1981	Richard Feynman	Proposal of quantum computers	Suggested using quantum systems to simulate physics
1985	David Deutsch	Universal quantum computer model	Established formal quantum computational framework
1994	Peter Shor	Shor's algorithm	Quantum factorization algorithm, foundation for quantum cryptography implications
1996	Lov Grover	Grover's search algorithm	Demonstrated quantum speed-up in data search
2001	IBM & Stanford University	First 5-qubit NMR-based quantum computer	Experimental proof of concept
2011	D-wave systems	First commercial 128-qubit quantum annealer	Practical step toward quantum hardware
2019	Google AI quantum	Quantum supremacy demonstrated	Performed a task faster than any classical computer
2023	IBM quantum	433-qubit osprey chip released	Largest superconducting processor to date

- The no-cloning theorem, which forbids copying unknown quantum states.
- A comparison between classical and quantum computing is shown in [Table 1](#).

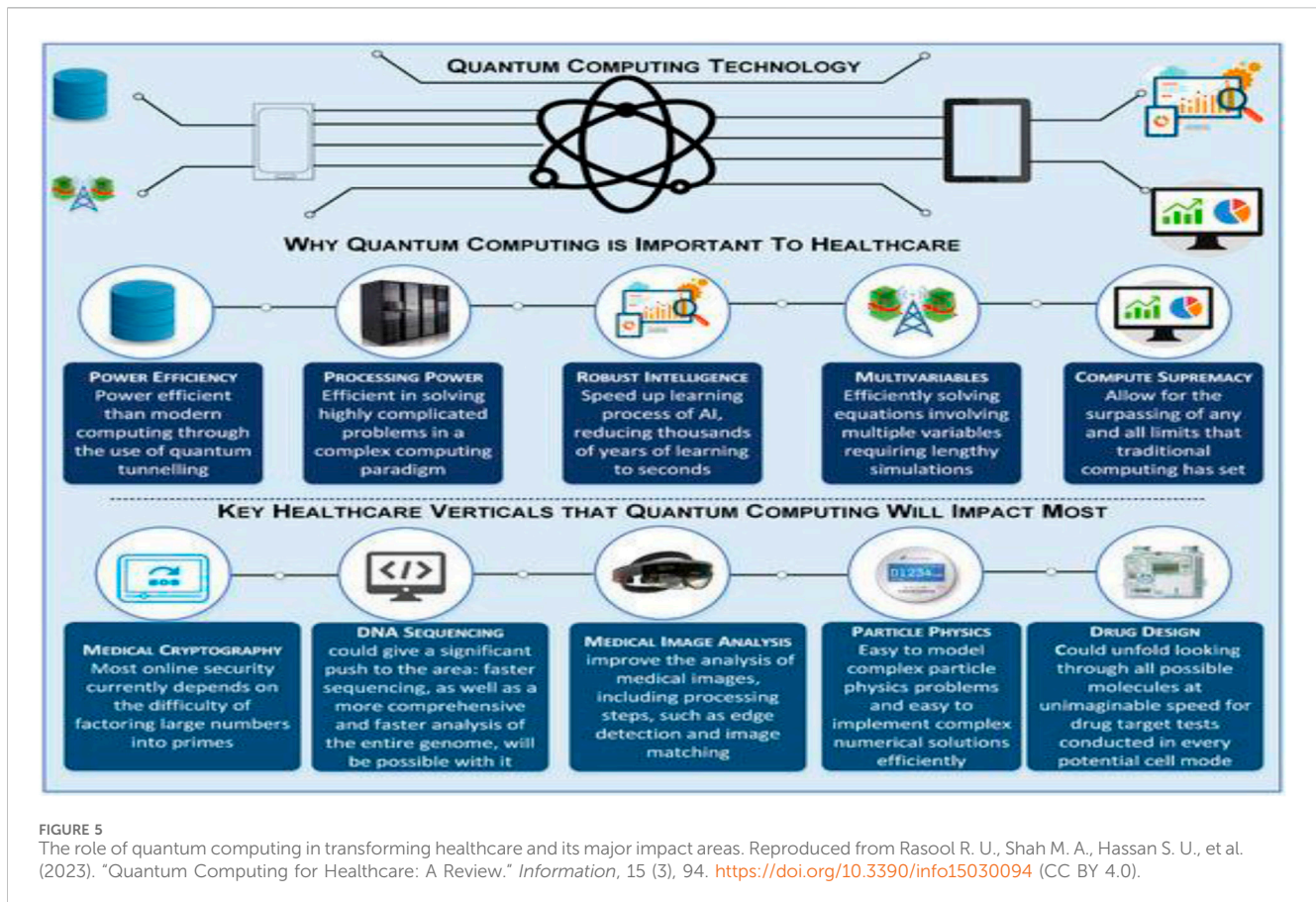
Current devices therefore belong to the *Noisy Intermediate-Scale Quantum (NISQ)* era ([Preskill, 2018](#); [Chae et al., 2024](#); [AbuGhanem, 2025](#)).

### 3.5 Qubits and quantum phenomena: from physics to practical impact

[Table 2](#) presents a chronological overview of the key milestones that shaped quantum computing, from early theoretical discoveries to present-day hardware advancements.

Quantum physical systems, such as the orientation of a photon or the spin of an electron, form the foundation for creating quantum bits (qubits). Quantum computers can thus exist in different configurations, including single-qubit systems ([Niel et al., 2010](#)), two-qubit systems ([Quantum, 2024](#)), and more complex multi-qubit architectures. A key historical milestone occurred in the early 2000s with the realization of the first five-qubit quantum processor ([Laflamme et al., 2001](#)). Since then, progress has accelerated rapidly; modern systems, such as IBM's Osprey chip, now feature up to 433 qubits ([IBM Quantum, 2023](#)).

Theoretical analyses suggest that achieving quantum supremacy—the point at which a programmable quantum computer can perform a task infeasible for classical computers—requires a minimum of roughly 50 entangled qubits ([Preskill, 2018](#); [Arute et al., 2019](#)). Quantum supremacy was first demonstrated when quantum devices performed specific random-



circuit sampling tasks faster than the world's most powerful supercomputers.

At a physical level, qubits mimic the quantum behavior of electrons orbiting an atomic nucleus, embodying three fundamental principles of quantum mechanics. The principles of superposition and entanglement, introduced earlier, manifest in physical systems that enable scalable quantum computation. A third key phenomenon—quantum interference—plays a vital role in amplifying correct computational outcomes and suppressing erroneous ones.

### 3.5.1 Quantum interference

At the subatomic level, particles exhibit wave-like behavior. When quantum states overlap, constructive interference amplifies the probability of correct computational paths, whereas destructive interference suppresses incorrect ones. This principle is central to algorithms such as Grover's search and Shor's factoring, where interference steers computation toward the optimal solution.

### 3.5.2 Applications and emerging frontiers

Quantum computing now spans diverse fields—cryptography, optimization, materials science, communication, and artificial intelligence. Recent practical algorithms leverage quantum principles to address complex real-world problems in finance, drug discovery, transportation, and weather prediction (illustrated in Figure 5).

A rapidly expanding area of application is disease diagnosis and healthcare. Quantum-enhanced machine-learning models have been used to analyze high-dimensional medical imaging and genomic datasets, accelerating the detection of diseases such as cancer, cardiovascular disorders, and neurodegenerative conditions. For example, quantum support-vector machines (QSVMs) and hybrid quantum-classical neural networks have shown promise in classifying MRI scans, predicting protein-folding patterns, and discovering biomarkers associated with complex diseases.

As hardware scalability and noise reduction improve, such applications are expected to mature from research prototypes to clinically reliable diagnostic tools. This underscores the broader vision of a physically scalable, fault-tolerant quantum paradigm—one capable of solving computationally intractable problems in healthcare and beyond.

## 4 Quantum algorithms

Quantum algorithms exploit quantum phenomena to achieve breakthroughs that are impossible or impractical with classical methods. Unlike classical algorithms that rely on binary logic, quantum algorithms leverage quantum mechanical properties to accelerate computation. Below are some of the most significant algorithms:

## 4.1 Shor's algorithm—integer factorization

Shor's algorithm, introduced in 1994, enables efficient factorization of large integers using the quantum Fourier transform (QFT). It reduces factoring to a periodicity problem, which quantum computers solve exponentially faster than classical ones. This poses a direct threat to RSA and other public-key cryptosystems. Its time complexity is polynomial, making it one of the clearest demonstrations of exponential quantum advantage, though it requires fault-tolerant quantum hardware with millions of physical qubits.

## 4.2 Grover's algorithm—search

Grover's algorithm provides a quadratic speedup for unstructured search problems. While classical search requires  $O(N)$  steps, Grover's algorithm solves it in  $O(\sqrt{N})$ . The algorithm amplifies the amplitude of the correct solution through iterative oracle queries and diffusion operations. Although the speedup is quadratic, its applications extend to optimization, pattern recognition, and cryptanalysis.

## 4.3 Quantum Fourier transform (QFT) and phase estimation (QPE)

QFT is a quantum analogue of the discrete Fourier transform, efficiently implemented in  $O(n^2)$  operations. It underpins Shor's algorithm and spectral estimation tasks. Quantum Phase Estimation (QPE) uses QFT to estimate eigenvalues of unitary operators, making it central to quantum chemistry and material science simulations.

## 4.4 Variational quantum algorithms (VQAs)

VQAs are hybrid algorithms designed for NISQ devices, combining quantum state preparation with classical optimization. The Variational Quantum Eigensolver (VQE) estimates ground-state energies in molecular systems, while the Quantum Approximate Optimization Algorithm (QAOA) addresses combinatorial optimization problems. These algorithms are practical for current noisy hardware, though they face challenges such as barren plateaus and noise sensitivity.

## 4.5 HHL algorithm—linear systems solver

The Harrow-Hassidim-Lloyd (HHL) algorithm solves systems of linear equations exponentially faster than classical methods under certain conditions (sparse, well-conditioned matrices). However, extracting full classical solutions diminishes this advantage. It remains most useful in quantum machine learning and scientific simulations where the solution can be consumed in quantum form.

## 4.6 NISQ era algorithms and quantum advantage

The current NISQ era is characterized by noisy devices with limited qubits. Algorithms such as VQE and QAOA are promising for near-term utility, while long-term advancements require scalable error correction. Early demonstrations of quantum supremacy show the field's potential, though practical, widely useful quantum advantage remains a frontier of research.

Early demonstrations of quantum supremacy have validated the potential of quantum algorithms such as Shor's and Grover's, but scalable fault-tolerant hardware remains a critical challenge.

Recent surveys also provide broader insights into the end-to-end complexity of quantum algorithms across diverse applications, highlighting practical limitations of quantum advantage. Dalzell (2023) present a comprehensive comparison of algorithmic primitives, performance metrics, and realistic hardware constraints. Complementarily, newer reviews such as *Quantum Algorithms for Quantum Molecular Systems: A Survey* (Lee et al., 2024) examine algorithmic adaptations for chemistry and molecular systems within fault-tolerant architectures.

Table 3 provides an integrated summary of the key quantum algorithms discussed above, including their core mechanisms, computational benefits, and hardware compatibility.

# 5 Applications of quantum

## 5.1 Cryptography

Quantum computing poses both significant risks and opportunities in modern cryptography.

- **Breaking classical encryption:** Shor's algorithm can factor large integers and compute discrete logarithms efficiently, directly threatening RSA, ECC, and other widely used public-key systems. A sufficiently powerful quantum computer could break RSA in hours, jeopardizing financial, governmental, and digital communication infrastructures.
- **Quantum-safe/Post-Quantum Cryptography (PQC):** To mitigate these risks, researchers are developing quantum-resistant algorithms, including lattice-based, code-based, hash-based, and multivariate schemes. Standardized PQC algorithms such as CRYSTALS-Kyber and CRYSTALS-Dilithium are emerging as practical replacements for vulnerable classical methods.
- **Quantum Key Distribution (QKD):** QKD protocols (BB84, E91, B92, SARG04, decoy-state QKD, MDI-QKD, CV-QKD) enable secure key establishment with information-theoretic security, as any eavesdropping introduces detectable disturbances.
- **Integration into optical networks:** Recent work investigates embedding QKD into existing telecom networks using models such as Key-as-a-Service (KaaS), enabling secure key distribution without fundamental changes to network infrastructure.
- **Other quantum cryptographic primitives:** Beyond QKD, quantum secure direct communication, quantum secret

**TABLE 3** Comprehensive summary of foundational and emerging quantum algorithms, highlighting their computational principles, performance advantages, limitations, and suitability across current NISQ devices and future fault-tolerant architectures.

Algorithm	Year/ developer	Core principle/ technique	Computational advantage	Primary applications	Limitations/ requirements	Hardware suitability
Shor’s algorithm	1994 – Peter Shor	Quantum Fourier transform (QFT) for integer factorization	<i>Exponential speedup</i> over classical factoring algorithms	Cryptography (breaking RSA, public-key systems)	Requires fault-tolerant hardware with millions of stable qubits	Fault-tolerant quantum computers (not feasible on NISQ)
Grover’s algorithm	1996 – Lov Grover	Amplitude amplification using oracle and diffusion operations	<i>Quadratic speedup</i> for unstructured search ( $O(\sqrt{N})$ vs. $O(N)$ )	Database search, optimization, pattern recognition, cryptanalysis	Limited to quadratic gain; requires multiple oracle evaluations	Can run on small-to medium-scale NISQ devices
Quantum fourier transform (QFT) and quantum phase estimation (QPE)	1994 – Developed within Shor’s framework	Fourier analysis and eigenvalue estimation on quantum states	Efficient $O(n^2)$ Fourier transform and spectral analysis	Quantum chemistry, materials modeling, molecular energy estimation	Highly sensitive to decoherence and rotation errors	Requires high-fidelity, low-noise qubits (beyond current NISQ)
Variational quantum algorithms (VQAs) — VQE and QAOA	2014 – Hybrid quantum-classical models	Parameterized quantum circuits with classical feedback optimization	<i>Near-term quantum advantage</i> for chemistry and optimization	Quantum chemistry (VQE), combinatorial optimization (QAOA), ML	Susceptible to barren plateaus and optimization instability	Best suited for NISQ-era devices
HHL algorithm	2009 – Harrow, Hassidim & Lloyd	Matrix inversion using quantum phase estimation	<i>Exponential speedup</i> for sparse, well-conditioned linear systems	Quantum ML, PDE solving, scientific simulations	Requires sparsity; limited utility when full classical readout needed.	Fault-tolerant quantum computers (future applicability)
NISQ era algorithms & quantum advantage	2016 – John Preskill et al.	Algorithms optimized for noisy, limited-qubit devices	Demonstrated <i>quantum utility</i> on current small systems	VQE, QAOA, random circuit sampling, quantum simulation	Limited by decoherence, gate errors, and low qubit counts	Specifically designed for NISQ hardware

**TABLE 4** Comparison of leading quantum computing hardware platforms in terms of qubit technology, scale, gate fidelity, coherence time, and quantum volume.

Platform	Technology	Qubit count	1Q gate fidelity	2Q gate fidelity	Coherence time (T1)	Quantum volume (QV)
IBM osprey	Superconducting	433	99.95%	98.5%	~100 $\mu$ s	512
IonQ Harmony	Trapped-ion	23	>99.9%	>99.0%	~10,000 $\mu$ s	32
Rigetti Aspen-M	Superconducting	80	~98.0%	~95.0%	~40 $\mu$ s	16
Google Sycamore	Superconducting	53	>99.5%	~99.0%	~100 $\mu$ s	128
Quantinuum H2	Trapped-ion	56	99.99%	99.7%	>10,000 $\mu$ s	2,048
PsiQuantum	Photonic (planned)	Projected 1M	TBD	TBD	TBD	TBD
Intel Horse Ridge	Silicon spin	~49	~98.5%	~92.0%	~100 $\mu$ s	TBD

The table reveals key distinctions: trapped-ion systems (IonQ, Quantinuum) excel in coherence time and gate fidelity, but generally lag in qubit scalability compared to superconducting platforms (IBM, Rigetti, Google). Quantinuum’s high QV, of 2048 reflects significant progress in algorithmic depth and circuit complexity. Photonic and silicon-spin qubits (PsiQuantum, Intel) are promising, though performance metrics remain under evaluation or in prototype stages.

These comparisons help contextualize platform readiness and guide use-case suitability for quantum machine learning, optimization, and secure cryptographic systems.

Sources: IBM, Research Blog (2024); Quantinuum Hardware Specs (2025); IonQ Technical Papers (2024); Rigetti Aspen-M, Datasheets (2023); Google Quantum Supremacy Paper; Intel Horse Ridge Project; PsiQuantum Roadmap (2025).

sharing, and quantum digital signatures extend quantum-secure communication capabilities. Their deployment depends on advances in detectors, single-photon sources, and quantum repeaters.

- Hybrid cryptographic systems: Transitional models combine classical cryptography, PQC, and QKD to strengthen security during the period before fault-tolerant quantum computers become widely available.

### 5.1.1 Post-quantum cryptography, QKD integration, and quantum-safe infrastructure

- Quantum-safe security has become a critical priority as quantum computers advance toward capabilities that can compromise classical encryption. A major research direction focuses on integrating Quantum Key Distribution (QKD) into existing optical network infrastructures. Frameworks such as Key-as-a-Service (KaaS) provide

TABLE 5 Comparison of major formal verification techniques for quantum programs.

Technique	Description	Tools/frameworks	Strengths	Limitations
Model checking	Represents quantum programs or protocols as transition systems; verifies correctness against temporal or probabilistic logic specifications.	QPMC (2025), PRISM (2025), and STORM (2025)	Automated; finds counterexamples; strong for correctness, safety, liveness; effective for small circuits.	State-explosion limits scalability; requires accurate models; limited to certain operations.
Deductive verification/proof systems	Uses logical frameworks, assertions, invariants, and theorem provers to formally prove algorithm correctness.	Qafny (2025), CoqQ (2025), QHLProver (2025), SMT solvers – Z3 (2025), PRISM (2025), and STORM (2025)	Can prove complex properties; human-readable proofs; strong for correctness and security guarantees.	Manual effort required, proof obligations can be large, difficult for very large programs; needs sophisticated math.
Compiler/circuit equivalence checking	Ensures transformations (optimizations, rewriting, compilation passes) preserve program semantics.	CertiQ (2025) and Quantum equivalence checker (2025)	Ensures compiled circuits are correct; supports safe optimization and simplification.	May not detect algorithm-level errors, limited to specific circuit classes, computationally expensive for large circuits.
Verification of error-correction/fault-tolerant programs	Validates quantum error correction routines and noise-resilient programs; checks encoded operations under realistic noise.	Veri-QEC (2025)	Critical for fault-tolerant quantum computing; ensures correct behavior of error-correction routines; handles non-ideal error models.	Complex error models, may rely on approximations or heuristics, proving guarantees under realistic noise is challenging.

practical deployment pathways, enabling secure key generation and distribution across current telecom networks (Aker et al., 2023). Parallel efforts aim to enhance QKD resilience through protocol optimization, improved security models, and advancements in supporting hardware including quantum repeaters, high-efficiency detectors, and stable single-photon sources (Sahu and Mazumdar, 2024; Zeydan, 2025).

- A growing consensus highlights the importance of hybrid classical-quantum cryptographic architectures. These approaches combine traditional cryptography, PQC algorithms, and QKD to provide layered protection during the transitional era in which quantum threats are emerging but not yet fully realized. Such hybrid systems are especially important for critical infrastructures—including Transportation Cyber-Physical Systems (TCPS), finance, and healthcare—where long-term data confidentiality is essential and vulnerability to quantum attacks is high (Chen and Tsai, 2025; Sahu and Mazumdar, 2024).
- An increasingly relevant threat is the “record-now, decrypt-later” strategy, where adversaries capture encrypted data today with the expectation of decrypting it once fault-tolerant quantum computers become available (Ott and Peikert, 2019). With recent advances demonstrating up to 48 stable logical qubits, the feasibility of quantum attacks on RSA and ECC is becoming more realistic (Rodriguez-Alvarez and Rodriguez-Merino, 2025). This underlines the urgency of immediate transition to quantum-secure solutions.
- Future research must focus on lightweight PQC designs, scalable key-management systems, and seamless integration frameworks that minimize computational overhead while maximizing security. Collectively, advancements in PQC, QKD, hybrid cryptosystems, and quantum-aware architectures form the foundation of a resilient quantum-secure cybersecurity ecosystem, preparing modern infrastructure for an era in which quantum capabilities continue to accelerate (Chhetri et al., 2025).

## 5.2 Applications of quantum computing in various domains

- Molecular simulation and quantum chemistry: Quantum computers can simulate molecules and reactions at the quantum level far more precisely than classical computers. This accelerates drug design (binding affinities, reaction pathways) and materials discovery (Benioff, 1980; Feynman, 1982; Dalzell, 2023; Chae et al., 2024).
- Quantum machine learning in medicine: Combining quantum and classical methods (QML) can help with genomic data processing, biomarker discovery, and patient stratification (Deutsch, 1985; Shor, 1995; Grover, 1996; Schuld, 2021).
- Clinical decision support and diagnostics: Quantum-enhanced algorithms may improve pattern recognition in imaging, faster Monte Carlo simulations for radiotherapy, and optimization of treatment planning (Brassard et al., 2000; Harrow et al., 2009).
- Emerging clinical applications: Recent literature further documents early real-world applications of quantum computing in medicine. For instance, a 2025 review in *Frontiers in Medicine* (Sharma et al., 2025) summarizes practical clinical use cases, including quantum-enhanced molecular modeling, radiotherapy planning, and medical image diagnostics. Such examples provide tangible evidence of progress from theoretical promise toward early-stage clinical deployment.

## 5.3 Artificial intelligence & machine learning

- Quantum-enhanced models: Quantum neural networks, quantum support vector machines, and hybrid quantum-classical architectures have been proposed to accelerate certain steps of training and inference (Harrow et al., 2009; Schuld, 2021).
- Feature space expansion and kernel methods: Quantum computers can map data into high-dimensional Hilbert

spaces more efficiently, assisting with kernel-based methods or inner-product computations in ML (Harrow et al., 2009; Schuld, 2021).

- Faster optimization and sampling: Quantum algorithms like quantum annealing, QAOA, or amplitude amplification may speed up optimization routines and probabilistic sampling tasks used in ML (e.g., in generative models) (Harrow et al., 2009; Farhi et al., 2014). Synergy in domains like drug discovery, finance, and logistics: AI + quantum computing together allow tackling tasks that are intractable for classical AI alone (e.g., large combinatorial optimization plus predictive modeling) (Dalzell, 2023; Herman, 2022).
- To provide a comprehensive view of progress in quantum machine learning (QML), Zaman et al. (2023) offer a recent survey detailing emerging architectures, hybrid models, and hardware–software integration challenges. Their study systematically categorizes QML approaches into quantum-enhanced, quantum-inspired, and hybrid frameworks, emphasizing how algorithmic efficiency depends on data encoding, circuit depth, and noise tolerance. The authors also discuss the rapid evolution of quantum kernel methods, variational models, and neural architectures, while highlighting open issues such as limited scalability, optimization instability, and the shortage of benchmark datasets. This work situates QML research within a broader context of algorithmic scalability and practical implementation barriers, aligning closely with the goals of this study. A comparison of current quantum hardware platforms is presented in Table 4.

## 5.4 Finance & business optimization

Portfolio optimization/asset allocation: Quantum (or quantum-inspired) optimization can evaluate many asset combinations in parallel to find more optimal returns vs. risk tradeoffs (Herman, 2022; IBM Quantum, 2025).

Derivative pricing, risk modeling, Monte Carlo simulations: Some quantum algorithms are suited for faster stochastic simulation and pricing of complex derivatives or measuring tail risks (VaR, CVaR) (Dalzell, 2023; Herman, 2022).

Credit scoring, fraud detection, loan risk assessment: Quantum machine learning methods may detect patterns in large data sets more efficiently in credit or fraud domains (Herman, 2022; Schuld, 2021).

Trading strategies, optimization for business operations and supply chains: Quantum approaches could optimize logistics, resource allocation, scheduling, route planning, and manufacturing. Several financial institutions are already experimenting with hybrid quantum-classical systems for trading or bond portfolio optimization (Herman, 2022; IBM Quantum, 2025).

## 5.5 Logistics & supply chains

Route optimization and vehicle routing problems: Quantum algorithms (e.g., QAOA, quantum annealing) can help solve large

combinatorial optimization tasks like the Traveling Salesman Problem (TSP), vehicle routing, or delivery scheduling more efficiently (Farhi et al., 2014; Dalzell, 2023).

Inventory management and demand forecasting: Quantum methods might improve forecasting models and optimize inventory across multiple warehouses and dynamic demand (Dalzell, 2023).

Transportation and traffic flow: Companies like Volkswagen and DHL have begun pilot experiments for quantum-assisted traffic planning and logistics routing (Dalzell, 2023; IBM Quantum, 2025). Recent studies have explored quantum cybersecurity, post-quantum cryptography, and reliability challenges in quantum software and infrastructures (Das et al., 2024; Ong et al., 2024; Baseri et al., 2024; Faruk et al., 2022; Majdoubi et al., 2024; Mamun et al., 2024; Moral, 2024; Robert, 2024; Scrivano, 2025; Sonko, 2024).

## 6 Verification and reliability in quantum programs

As quantum hardware and algorithms scale in complexity, ensuring correctness and reliability becomes ever more critical. Quantum programs are fundamentally harder to test and debug than classical ones because of probabilistic measurement outcomes, noise and errors in hardware/software, and limited resources in NISQ (Noisy Intermediate-Scale Quantum) devices. Formal verification is one of the most promising ways to provide guarantees about quantum software correctness (Lewis et al., 2021; Shi, 2019; Bauer-Marquart et al., 2022; Gill et al., 2020). A comparison of major formal verification techniques for quantum programs, including their strengths and limitations, is presented in Table 5.

### 6.1 Key challenges

Probabilistic Outcomes & Measurement Collapse: Quantum algorithms often produce probability distributions over outputs; small errors or noise can lead to wrong distributions, and measurement itself collapses superposition (Lewis et al., 2021; Shi, 2019).

Noise, Decoherence & Gate Errors: Real quantum hardware is imperfect. Errors can come from limited coherence times, imperfect gate operations, cross-talk, and measurement error (Preskill, 2018; Shi, 2019).

Resource Constraints in NISQ Devices: Limited qubit count, limited circuit depth, high error rates restrict how large/complex programs can be, which makes verification more difficult (Preskill, 2018; Lewis et al., 2021).

Complex State Space: Even a modest number of qubits gives a huge state space (dimension grows exponentially), which complicates exhaustive reasoning or simulation (Preskill, 2018; Lewis et al., 2021).

### 6.2 Formal verification approaches

To address these challenges, researchers have developed several formal methods adapted to quantum programs:

## 6.3 Verification workflow & best practices

**Specify precisely:** Write formal preconditions, postconditions, invariants, and expected behavior. Clear specification is essential (Qafny, 2025; CoqQ, 2025; Veri-QEC, 2025).

**Modular design:** Break large quantum programs or circuits into smaller, verifiable components (subroutines, logical qubits, etc.) (Qafny, 2025; CoqQ, 2025; Veri-QEC, 2025).

**Use abstraction and compositionality:** Abstractions can reduce complexity; compositional verification allows reasoning about parts separately (Qafny, 2025; CoqQ, 2025; Veri-QEC, 2025).

**Combine automated tools with interactive proofs:** Some parts may be checked automatically (SMT solvers, model checkers), others need manual proof or human insight (QPMC, 2025; PRISM, 2025; STORM, 2025; Qafny, 2025; CoqQ, 2025; QHLProver, 2025; SMT solvers–Z3, 2025).

**Validate error models:** The verification must assume realistic noise and error models; if the model is too optimistic, real-world reliability might differ (Veri-QEC, 2025).

**Testing and simulation:** Complement formal verification with simulations, randomized benchmarking, and hardware tests to cross-validate (Veri-QEC, 2025).

## 7 Quantum computing in finance and machine learning

### 7.1 Finance

**Quantum Amplitude Estimation (QAE) for derivative pricing, risk analysis, portfolio valuation:** A prime example is “Quantum Portfolio Value Forecasting,” where amplitude estimation is used to estimate the expected long-term value of a portfolio using a variant of the Gordon-Shapiro formula. The authors apply it to a 5-asset portfolio on actual trapped-ion quantum hardware, showing that for the same computational cost one can reduce statistical error compared to classical estimates (Herman, 2022; Schuld, 2021).

Also, “Quantum advantage for multi-option portfolio pricing and valuation adjustments (CVAs)” studies how quantum Monte Carlo methods can accelerate estimation of CVAs and similar risk-adjusted derivative valuations (Herman, 2022; Schuld, 2021).

**Portfolio Optimization using QAOA & Higher-Order Moments:** A recent work “Higher-Order Portfolio Optimization with Quantum Approximate Optimization Algorithm” (Uotila et al., 2025) formulates portfolio optimization including third and fourth moments (skewness, kurtosis). The problem becomes a higher-order unconstrained binary optimization (HUBO) rather than a quadratic one, and QAOA is used to optimize it (Herman, 2022; Schuld, 2021). Their experiments (on many test instances) show that including higher moments yields better portfolio allocations than classical baselines under certain settings.

Another example is “Enhancing Knapsack-based Financial Portfolio Optimization Using QAOA” (Huot et al., 2024), which frames the portfolio problem as a knapsack problem and uses QAOA with quantum walk mixers (Herman, 2022; Schuld, 2021). They show competitive approximate ratios for certain asset-selection constraints.

**Challenges and realistic constraints:** While quantum methods promise quadratic speedup in many Monte Carlo tasks (pricing, risk, VaR, CVaR), actual hardware, noise, and limited qubit counts often limit demonstration of that speedup. Furthermore, designing cost Hamiltonians, encoding classical data, and managing readout error are key issues. Surveys (e.g., Herman, 2022) provide comparisons of different algorithmic methods and highlight that many financial use cases are still in simulation or at small scale (Herman, 2022; Schuld, 2021).

### 7.2 Machine learning

**Quantum Kernel Methods & Quantum Neural Networks (QNNs):** Recent benchmarking studies (e.g., “Quantum kernel methods under scrutiny: a benchmarking study,” 2025) compare different quantum kernel variants (fidelity quantum kernels, projected quantum kernels) across many datasets for both classification and regression (Schuld, 2021).

Another important result is from “Supervised quantum machine learning models are kernel methods” by Schuld (2021). This work argues that many quantum models that are called “quantum neural networks” have mathematical structures very similar to kernel methods (they embed data into a high-dimensional Hilbert space via encoding circuits, then perform learning via overlaps/inner products of those embeddings) (Schuld, 2021).

**Applications, robustness, and adversarial aspects:** Studies like “Benchmarking Adversarially Robust Quantum Machine Learning at Scale” (West, Erfani, Leckie, et al., 2022) show that quantum variational classifiers (QVCs) can be more robust to classical adversarial attacks in certain settings, possibly owing to different features learned in quantum circuits (Schuld, 2021).

**Current Trends & Limitations:** There is growing interest in combining quantum methods with classical ML pipelines, such as feature embedding + kernel learning, hybrid QNNs, federated quantum neural networks, and quantum boosted models. However, limitations remain:

The overhead of encoding classical data into quantum states (Schuld, 2021).

The effect of noise and error accumulation (especially for deep circuits) (Schuld, 2021).

The question of when quantum methods actually outperform classical ones on real, large, high-dimensional data, not just toy or synthetic datasets (Schuld, 2021).

Benchmarking across data sets, comparing run-time, sampling cost, and hardware-noise trade-offs; reproducibility is also a concern (Schuld, 2021).

### 7.3 Hardware benchmarking

To address the need for quantitative assessment of quantum computing platforms, the following table provides a comparative overview of leading hardware systems as of 2024–2025. This includes metrics such as qubit count, gate fidelities, coherence times, and quantum volume.

## 8 Challenges and limitations

While quantum computing carries immense potential, several fundamental and engineering challenges must be overcome before practical, large-scale quantum computers become commonplace.

### 8.1 Quantum decoherence

Qubits are extremely fragile. Even minor interactions with the surrounding environment (electromagnetic noise, thermal fluctuations, stray photons, vibrations) can cause them to lose coherence—that is, the phase relationships between amplitude states collapse, turning quantum superpositions into classical mixtures. This process is known as decoherence (Preskill, 2018; Arute et al., 2019; Dalzell, 2023).

Decoherence places strict time limits on how long quantum computations can run before errors overwhelm the system. Mitigation techniques like dynamical decoupling, decoherence-free subspaces, and improved shielding help, but they do not fully eliminate the problem (Preskill, 2018; Dalzell, 2023).

### 8.2 Error correction and fault tolerance

Because qubits are noisy and imperfect, Quantum Error Correction (QEC) is essential. However, implementing QEC typically requires many physical qubits per logical qubit—sometimes hundreds or even thousands—just to protect one “ideal” qubit (Preskill, 2018; Dalzell, 2023; Chae et al., 2024).

The overhead in gates, measurement, syndrome extraction, and classical control is substantial. Achieving fault tolerance (meaning the system can continue correct operation despite errors) remains a grand engineering challenge (Preskill, 2018; Dalzell, 2023).

### 8.3 High cost & physical requirements

Many quantum platforms require extreme operating conditions—ultra-low temperatures (millikelvin via dilution refrigerators), ultra-high vacuum, electromagnetic shielding, cryogenic control electronics, and complex calibration systems (Preskill, 2018; IBM Quantum, 2025).

Moreover, cooling, isolation, and control infrastructure themselves consume energy and space, making the entire system bulky and costly (Preskill, 2018; IBM Quantum, 2025).

### 8.4 Limited qubit count and connectivity

Currently, even the most advanced quantum systems remain limited to hundreds or a few thousand physical qubits (or less) with nonideal connectivity (Preskill, 2018; IBM Quantum, 2025).

To solve real-world, large-scale problems, the field will likely require millions of qubits (or many logical qubits after error correction). Achieving that level of scaling, with high-fidelity inter-qubit operations and low cross-talk, is a central bottleneck (Preskill, 2018; IBM Quantum, 2025).

## 8.5 Verification, validation & debugging complexity

Verifying the correctness of large quantum programs is extremely challenging. The exponential state space, probabilistic outputs, and noise make classical simulation infeasible beyond modest sizes. Formal verification (via model checking, deductive proofs, equivalence checking) is still in early stages (Lewis et al., 2021; Gill et al., 2020).

Furthermore, debugging quantum circuits is nontrivial—observing intermediate states collapses them, and errors may manifest subtly due to interference effects (Lewis et al., 2021; Gill et al., 2020).

A recent survey by Hiremath et al. (2025) classifies quantum computing challenges into short-term (hardware noise, scalability, resource overhead) and long-term (fault tolerance, architecture standardization, and interdisciplinary integration). Incorporating such frameworks helps delineate the layered nature of current and future quantum computing challenges.

## 9 Current status and future outlook

### 9.1 Leading players & roadmaps

IBM has published an explicit roadmap to progress toward fault-tolerant quantum computers, targeting hardware improvements and modular systems through 2033 (IBM Quantum, 2025).

Google maintains its Quantum AI roadmap, aiming to push hardware and algorithms forward toward practical quantum advantage applications (IBM Quantum, 2025).

Microsoft, Intel, Amazon are investing heavily in quantum ecosystems—development tools, cloud platforms, qubit research, and software stacks (Preskill, 2018; Dalzell, 2023).

Rigetti, IonQ, PsiQuantum, others are active in hardware innovation. For example, IonQ’s roadmap includes scaling to many thousands of qubits in coming years (Preskill, 2018; IBM Quantum, 2025).

### 9.2 Trends & predictions

Experts often forecast that quantum computers will initially complement classical systems, rather than fully replace them, targeting niche areas where quantum advantage emerges (Preskill, 2018; Dalzell, 2023).

Some believe within the next decade, we may see “quantum utility”—device-scale applications where quantum methods give practical gains in specific domains (e.g., chemistry, optimization) (Preskill, 2018; Dalzell, 2023).

Beyond that, fault-tolerant universal quantum computing may arrive in ~15–20 years (or more), depending on engineering progress (Preskill, 2018; Dalzell, 2023).

Predictions are cautious: many believe scaling, error correction, verification, and cost are still vast hurdles before broad deployment (Preskill, 2018; Dalzell, 2023).

## 10 Conclusion

Quantum computing is not just another incremental technology—it represents a paradigm shift, enabling new computational capabilities rooted in quantum mechanics (Benioff, 1980; Feynman, 1982; Deutsch, 1985; Shor, 1995; Grover, 1996; Preskill, 2018; Dalzell, 2023). Its potential spans cryptography, materials science, drug discovery, optimization, AI, finance, and beyond. Yet substantial challenges remain: decoherence, error correction, scaling, environmental and physical constraints, and software reliability (Preskill, 2018; Dalzell, 2023; Chae et al., 2024; IBM Quantum, 2025).

While the promise is vast, the path is hard. The coming years will likely see hybrid systems—quantum + classical—co-operating to solve tasks beyond reach today (Preskill, 2018; Dalzell, 2023). Success will require co-design across physics, hardware, algorithms, and software engineering, along with rigorous benchmarking and verification (Lewis et al., 2021; Gill et al., 2020). As scientists, engineers, and institutions continue driving progress, we inch closer to unleashing quantum's full power and transforming what we regard as computationally possible (Benioff, 1980; Feynman, 1982; Deutsch, 1985; Shor, 1995; Grover, 1996; Preskill, 2018; Dalzell, 2023).

The inclusion of recent studies (Dalzell, 2023; Zaman et al., 2023; Hiremath et al., 2025; Sharma et al., 2025) strengthens the contextual grounding of this paper, ensuring that the discussion reflects current advancements and the most recent perspectives in quantum algorithm design, medical applications, and hardware scalability.

## Author contributions

VR: Writing – review and editing.

## References

- AbuGhanem, M. (2025). Superconducting quantum computers: who is leading the future? *EPJ Quantum Technol.* 12, 102. doi:10.1140/epjqt/s40507-025-00405-7
- Akter, M. S., Rodriguez-Cardenas, J., Shahriar, H., Cuzzocrea, A., and Wu, F. (2023). Quantum cryptography for enhanced network security: a comprehensive survey of research, developments, and future directions. *IEEE Big Data* 2021, 5408–5417. doi:10.1109/bigdata59044.2023.10386889
- Andersen, L., Berta, P., and Reberstrost, C. (2025). Hardware–software co-design for quantum advantage: bridging algorithms and architectures. *Nat. Rev. Phys.* 7, 450–464. doi:10.1038/s42254-025-00918-z
- Arute, F., Arya, K., Babbush, R., Bacon, D., Bardin, J. C., Barends, R., et al. (2019). Quantum supremacy using a programmable superconducting processor. *Nature* 574, 505–510. doi:10.1038/s41586-019-1666-5
- Baseri, Y., Chouhan, V., and Hafid, A. (2024). Navigating quantum security risks in networked environments: a comprehensive study of quantum-safe network protocols. *Comput. Secur.* 142, 103883. doi:10.1016/j.cose.2024.103883
- Bauer-Marquart, F., Leue, S., and Schilling, C. (2022). symQV: automated symbolic verification of quantum programs. arXiv:2212.02267.
- Benioff, P. (1980). The computer as a physical system: a microscopic quantum mechanical Hamiltonian model of computers as represented by turing machines. *J. Stat. Phys.* 22, 563–591. doi:10.1007/BF01011339
- Brassard, G., Høyer, P., Mosca, M., and Tapp, A. (2000). Quantum amplitude amplification and estimation. arXiv:quant-ph/0005055.
- Carter, M., and Gheorghiu, E. (2024). QBench and QPack: frameworks for benchmarking quantum algorithms. *ACM Trans. Quantum Comput.* 6 (3). doi:10.1145/3691041
- CertiQ. (2025) CertiQ – compiler/circuit equivalence verification (arXiv).
- Chae, E., Choi, J., and Kim, J. (2024). An elementary review on basic principles and developments of qubits for quantum computing. *Nano Converg.* 11, 11. doi:10.1186/s40580-024-00418-5
- Chen, S.-J., and Tsai, Y. (2025). Quantum-safe networks for 6G. 2, 1. doi:10.69709/caic.2025.102135
- Chen, J.-S., Nielsen, E., Ebert, M., Inlek, V., Wright, K., Chaplin, V., et al. (2024). Benchmarking a trapped-ion quantum computer with 30 qubits. *Quantum* 8, 1516. doi:10.22331/q-2024-11-07-1516
- Chhetri, G., Somvanshi, S., Hebli, P., Brotee, S., and Das, S. (2025). Post-quantum cryptography and quantum-safe security. arXiv:2510.10436.
- Chohan, A. (2024). A comparative review of quantum bits: superconducting, topological, spin, and emerging qubit technologies. *SSRN Prepr.* Available online at: [https://papers.ssrn.com/sol3/papers.cfm?abstract\\_id=4979773](https://papers.ssrn.com/sol3/papers.cfm?abstract_id=4979773).
- Colobridge Blog (2025). Quantum computing 2025 — comparison of leading Qubit technologies. Available online at: [https://blog.colobridge.net/wp-content/uploads/2025/09/Comparison-of-Leading-Qubit-Technologies\\_%D0%B0%D0%BD%D0%B3%D0%BB-1-1024x536.jpg](https://blog.colobridge.net/wp-content/uploads/2025/09/Comparison-of-Leading-Qubit-Technologies_%D0%B0%D0%BD%D0%B3%D0%BB-1-1024x536.jpg).
- CoqQ. (2025) CoqQ – deductive verification framework for quantum programs (arXiv).
- Dalzell, M. (2023). *Quantum algorithms: a survey of applications and end-to-end complexities*. Cambridge, UK: Cambridge University Press.
- Das, A., Singh, T., and Kumar, P. (2024). QSEC: Quantum software error correction and certification framework. *IEEE Trans. Quantum Eng.* 5, 5203012. doi:10.1109/TQE.2024.5203012
- Deutsch, D. (1985). Quantum theory, the church–turing principle and the universal quantum computer. *Proc. R. Soc. A* 400, 97–117.

## Funding

The author(s) declared that financial support was not received for this work and/or its publication.

## Conflict of interest

The author(s) declared that this work was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

## Generative AI statement

The author(s) declared that generative AI was used in the creation of this manuscript. ChatGPT.

Any alternative text (alt text) provided alongside figures in this article has been generated by Frontiers with the support of artificial intelligence and reasonable efforts have been made to ensure accuracy, including review by the authors wherever possible. If you identify any issues, please contact us.

## Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

- EPJ Quantum Technology (2021). Industry quantum computing applications: bridging algorithms and hardware. Available online at: <https://epjquantumtechnology.springeropen.com/articles/10.1140/epjqt/s40507-021-00114-x>.
- Farhi, E., Goldstone, J., and Gutmann, S. (2014). A quantum approximate optimization algorithm. arXiv:1411.4028.
- Faruk, M. J. H., Tahora, S., Tasnim, M., Shahriar, H., and Sakib, N. (2022). A review of quantum cybersecurity. doi:10.1109/icaic53980.2022.9896970
- Feynman, R. P. (1982). Simulating physics with computers. *Int. J. Theor. Phys.* 21, 467–488. doi:10.1007/bf02650179
- Gill, S. S., Kumar, A., Singh, M., Kaur, K., Buyya, R., Usman, M., et al. (2020). Quantum computing: a taxonomy, systematic review and future directions. *Softw. Pract. Exp.* 50 (6), 1074–1106. doi:10.1002/spe.3039
- Goswami, S., and Patel, P. (2025). Hybrid qubit systems: integrating photonic, NV-center and superconducting technologies. *J. Quantum Eng.* 3. doi:10.1088/2633-4356/ad19d2
- Grigoryan, A., Kumar, S., and Pinheiro, P. R. (2025). A review on models and applications of quantum computing. *Computation* 7 (3), 39. doi:10.3390/computation7030039
- Grover, L. K. (1996). “A fast quantum mechanical algorithm for database search,” in *Proceedings of STOC, 1996*. arXiv:quant-ph/9605043.
- Harrow, W., Hassidim, A., and Lloyd, S. (2009). Quantum algorithm for linear systems of equations. *Phys. Rev. Lett.* 103, 150502. doi:10.1103/PhysRevLett.103.150502
- Herman, D. (2022). A survey of quantum computing for finance. arXiv:2201.02773.
- Hiremath, S., Kumar, P., and Singh, R. (2025). A literature survey on quantum computing in next generation challenges in circuit design and applications of future enabling technologies. *Int. J. Comput. Appl.* 187 (1), 1–8.
- Huot, C., Heng, S. V., Kim, T.-K., and Han, Y. (2024). Quantum autoencoder for enhanced fraud detection in imbalanced credit card dataset. *IEEE Access* 12, 169671–169681. doi:10.1109/ACCESS.2024.3496901
- Huynh, L., Hong, J., Mian, A., Suzuki, H., Wu, Y., and Camtepe, S. (2023). Quantum-inspired machine learning: a survey. arXiv preprint arXiv:2308.11269.
- IBM Quantum (2023). *IBM osprey: 433-qubit quantum processor*. IBM Research Blog. Available online at: <https://research.ibm.com/blog/ibm-osprey>.
- IBM Quantum (2025). “IBM Quantum Roadmap” (online). (IBM roadmap and 2024–2025 updates).
- Khabiboulline, E., Romero, J., and Kubica, A. (2025). Bridging theory and implementation in quantum computing: challenges and opportunities. *Commun. ACM* 68 (7), 72–81. doi:10.1145/3679312
- Kim, D., Yamamoto, Y., and Gill, S. (2025). Photonic and hybrid quantum processors: integration challenges and opportunities. *IEEE Trans. Quantum Eng.* 6, 5402311. doi:10.1109/TQE.2025.5402311
- Kumar, A. (2022). A brief history of quantum computing. *Medium*. Available online at: [https://miro.medium.com/v2/resize:fit:1400/1\\*N3CWuTrxr-tg2712\\_9Xrmw.png](https://miro.medium.com/v2/resize:fit:1400/1*N3CWuTrxr-tg2712_9Xrmw.png).
- Laflamme, R., Knill, E., and Zurek, W. H. (2001). *Demonstration of a five-qubit quantum computer*. IBM Research News Archive, IBM Research.
- Lee, S., Patel, D., and Wang, Y. (2024). Quantum algorithms for quantum molecular systems: a survey. *WIREs Comput. Mol. Sci.* 15 (2), e70020. doi:10.1002/wcms.70020
- Lewis, M., Soudjani, S., and Zuliani, P. (2021). Formal verification of quantum programs: theory, tools and challenges. arXiv:2110.01320.
- Majdoubi, C., Mendili, S. E., and Gahi, Y. (2024). Quantum cryptology in the big data security era. *Int. J. Adv. Comput. Sci. Appl.* 15. doi:10.14569/ijacsa.2024.0150761
- Mamun, A. A., Abrar, A., Rahman, M., Salek, M. S., and Chowdhury, M. (2024). Enhancing TCPS security: a shift to PQC. arXiv:2411.13023.
- Monroe, T. (2024). Recent advances in trapped-ion quantum processors. *Nat. Photonics* 18, 210–223. doi:10.1038/s41566-024-01987-y
- Moral, J. O. (2024). Cybersecurity in critical infrastructures: a PQC perspective.
- Morales, S., Leue, E., and Bauer-Marquart, F. (2025). Hybrid-QEC: integrating classical and quantum verification for fault-tolerant compilation. *npj Quantum Inf.* 11 (33), 1–14. doi:10.1038/s41534-025-00832-y
- Nielsen, M. A., and Chuang, I. L. (2010). *Quantum computation and quantum information*. 10th Anniversary Edition. Cambridge: Cambridge University Press.
- Ong, M., Kwon, J., and Schilling, C. (2024). Survey of reliability challenges in quantum software stacks: testing, debugging, and verification. arXiv preprint arXiv:2409.06741.
- Ott, D. J., and Peikert, C. (2019). Post quantum cryptography migration. arXiv:1909.07353.
- Patel, M., and Zurek, R. (2025). Integrated quantum hardware–software co-design: a roadmap toward practical quantum advantage. *IEEE Access* 13, 78512–78528. doi:10.1109/ACCESS.2025.3459821
- Peruzzo, A., McClean, J., Shadbolt, P., Yung, M. H., Zhou, X. Q., Love, P. J., et al. (2014). A variational eigenvalue solver on a photonic quantum processor. *Nat. Commun.* 5, 4213. doi:10.1038/ncomms5213
- Preskill, J. (2018). Quantum computing in the NISQ era and beyond. *Quantum* 2, 79. doi:10.22331/q-2018-08-06-79
- PRISM (2025). PRISM – probabilistic model checking tool (ACM Digital Library).
- Qafny (2025). Qafny – proof system for quantum programs (arXiv).
- QEC (2025). Quantum equivalence checker – circuit equivalence tool (dcs.gla.ac.uk).
- QHLProver (2025). QHLProver – Quantum hoare logic prover (arXiv).
- QPMC (2025). QPMC – quantum program/protocol model checker (opus.lib.uts.edu.au).
- Quantum, I. B. M. (2024). Understanding multi-qubit interactions and two-qubit gates. in *IBM quantum learning*. Available online at: <https://quantum-computing.ibm.com>.
- Quantum, I. B. M. (2025). *IBM quantum system two architecture and modular scaling approach*. IBM Research Blog. Available online at: <https://research.ibm.com/blog/ibm-quantum-system-two>.
- Quantum Algorithm Zoo (2025). Comprehensive catalogue of quantum algorithms. Available online at: <https://quantumalgorithmzoo.org/>.
- Quantum Computing (2021). *Pioneers of quantum computing*. SlideShare Presentation. Available online at: <https://image.slidesharecdn.com/quantumcomputing-211118090146/85/quantum-computing-5-320.jpg>.
- Reynolds, D., Bravyi, T., and Lloyd, S. (2025). The next decade of quantum computing: from concept to scalable systems. *Nat. Rev. Phys.* 7, 530–547. doi:10.1038/s42254-025-00943-y
- Robert, W. (2024). Cryptographic techniques for IoMT security.
- Rodriguez-Alvarez, N., and Rodriguez-Merino, F. (2025). Performance and storage analysis of CRYSTALS-Kyber.
- Sahu, S. K., and Mazumdar, K. (2024). Analysis of quantum cryptography applications.
- Schuld, M. (2021). *Supervised quantum machine learning models are kernel methods*. arXiv:2101.11020.
- Scrivano, A. (2025). Comparative study of classical and post-quantum algorithms.
- Sharma, N., Verma, R., Patel, T., Wilkes, B. G., and Panuganti, B. (2025). Applications of quantum computing in clinical care. *Front. Med.* 12, 1573016. doi:10.3389/fmed.2025.1573016
- Shi, Y. (2019). CertiQ: a mostly-automated verification of a realistic quantum compiler. arXiv:1908.08963.
- Shor, P. W. (1995). Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. arXiv:quant-ph/9508027.
- SMT solvers – Z3, (n.d.). Yices (ACM digital Library).
- Sonko, S. (2024). Quantum cryptography and U.S. digital security.
- STORM (2025). STORM – probabilistic model checking tool (ACM Digital Library).
- Uotila, V., Ripatti, J., and Zhao, B. (2025). Higher-order portfolio optimization with quantum approximate optimization algorithm. arXiv preprint arXiv:2509.01496.
- Veri-QEC (2025) Veri-QEC – verification framework for quantum error-correcting programs (arXiv).
- Vidal, T., Roser, F., and Lewis, M. (2025). Quantum optimization for finance and logistics: benchmarking real-world feasibility. *npj Quantum Inf.* 11 (27). doi:10.1038/s41534-025-00790-4
- Zaman, F., Ali, S., Hussain, M., and Khan, A. (2023). A survey on quantum machine learning: current trends, challenges, opportunities, and the road ahead. arXiv preprint arXiv:2310.10315.
- Zeydan, E. (2025). Quantum technologies for beyond 5G and 6G networks.
- Zhang, A., Kim, S., and Haataja, M. P. (2024). Quantum chemistry algorithms under noise: resource estimation and scalability. *J. Chem. Phys.* 160 (14), 244903. doi:10.1063/5.0198773
- Zhao, Q., and Kumar, M. (2024). Benchmarking and noise characterization for heterogeneous quantum hardware. *Phys. Rev. Appl.* 21 (4), 045012. doi:10.1103/PhysRevApplied.21.045012
- Zhou, R., Gheorghiu, M., and Brown, K. (2024). QuVerify: a scalable framework for end-to-end verification of quantum programs. *ACM Trans. Quantum Comput.* 6 (2), 1–22. doi:10.1145/3689127
- Zhu, L., and Browne, K. (2025). Topological qubits and quantum error protection: a review. *Nat. Rev. Phys.* 7, 615–630. doi:10.1038/s42254-025-00976-4