Expert Systems With Applications 194 (2022) 116512

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

Review

Machine learning in the quantum realm: The state-of-the-art, challenges, and future vision

Essam H. Houssein [a,*], Zainab Abohashima [b], Mohamed Elhoseny [c,d], Waleed M. Mohamed [a]

[a] Faculty of Computers and Information, Minia University, Minia, Egypt
[b] Faculty of Computer Science, Nahda University, Beni-Suef, Egypt
[c] University of Sharjah, Sharjah, United Arab Emirates
[d] Faculty of Computers and Information, Mansoura University, Egypt

ARTICLE INFO

Keywords:
Quantum machine learning
Quantum computing
Quantum deep learning
Quantum inspired
Hybrid quantum–classical
Quantum classification
Variational quantum algorithms

ABSTRACT

Machine learning has become a ubiquitous and effective technique for data processing and classification. Furthermore, due to the superiority and progress of quantum computing in many areas (e.g., cryptography, machine learning, healthcare), a combination of classical machine learning and quantum information processing has established a new field, called, quantum machine learning. One of the most frequently used applications of quantum computing is machine learning. This paper aims to present a comprehensive review of state-of-the-art advances in quantum machine learning. Besides, this paper outlines recent works on different architectures of quantum deep learning, and illustrates classification tasks in the quantum domain as well as encoding methods and quantum subroutines. Furthermore, this paper examines how the concept of quantum computing enhances classical machine learning. Two methods for improving the performance of classical machine learning are presented. Finally, this work provides a general review of challenges and the future vision of quantum machine learning.

1. Introduction

Machine learning (ML) is one of the more well-established and widespread sub-field in artificial intelligence and computer science. ML algorithms can be classified into three main algorithms: supervised learning algorithms, unsupervised learning algorithms, and reinforcement learning algorithms. Firstly, a supervised learning algorithm can train a classifier to known input and output data to predict new data (e.g., classification and regression techniques) (Jiang, Gradus, & Rosellini, 2020). Secondly, an unsupervised learning algorithm can find hidden patterns in data (e.g., clustering techniques) (Sinaga & Yang, 2020). Finally, the reinforcement learning algorithms can learn from experiences and detect the best actions from an unseen environment to achieve optimal state transition for obtaining the goal (Botvinick et al., 2019; Coronato, Naeem, De Pietro, & Paragliola, 2020). ML is used in various applications (i.e., healthcare, cryptography, and pattern classification) and often achieves higher accuracy and performance without human decisions. Due to the enormous increase in various data types (i.e., images, text, videos, and record audios), challenges have arisen in machine learning (i.e., high-cost learning and kernel estimation) (Kubat, 2017; Mehta et al., 2019).

Deep learning (DL) is currently one of the most popular research topics in the field of ML. DL is a branch of ML that involves several layers, such as a neural networks (NNs). DL can automatically learn representations from huge data, such as images, text, video, and audio, without human knowledge. DL learns the hidden structure lies within the collected data, transfer data across multiple layers, and each layer can eventually extract features and transfer them to the next layer (LeCun, Bengio, & Hinton, 2015). There are two main DL architectures: convolutional neural networks (CNNs) and recurrent neural networks (RNNs). DL plays a critical role in computer vision applications, such as image classification, image segmentation, and text classification.

The last 10 years have witnessed striking research in quantum computing (QC). Fig. 1a presents statistics on QC from 2010 to 2020. Besides, Fig. 1b presents the distribution of QC in different areas according to the Scopus database. QC has been developed and has demonstrated superior performance compared to its counterpart, classical computing. QC uses features of QM, such as superposition, interference, and entanglement for processing information, and is used in various applications, such as cryptography (Broadbent & Schaffner,

* Corresponding author.
E-mail addresses: essam.halim@mu.edu.eg (E.H. Houssein), zeinab.saad@nub.edu.eg (Z. Abohashima), melhoseny@ieee.org (M. Elhoseny), waleedmakram@minia.edu.eg (W.M. Mohamed).

https://doi.org/10.1016/j.eswa.2022.116512
Received 16 December 2020; Received in revised form 29 December 2021; Accepted 4 January 2022
Available online 21 January 2022
0957-4174/

2016), artificial intelligence (Dunjko & Briegel, 2018), and healthcare (Cao, Romero, & Aspuru-Guzik, 2018; Li, Zhou, Xu, Luo and Jiang, 2020). Different algorithms and approaches have been published in the ML field based on QC laws, called, quantum machine learning (QML) (Biamonte et al., 2017; Schuld, Sinayskiy, & Petruccione, 2015a).

In this work, QML algorithms in the quantum realm are basically classified into three paradigms. In 2014, Rebentrost, Mohseni, and Lloyd (2014) presented a quantum version of the support vector machine (QSVM) for the classification of big data, which is the first purely QML paradigm. QSVM achieved a logarithmic speedup over classical counterparts. The second paradigm is quantum-inspired ML. Sergioli et al. proposed a binary quantum classifier inspired by the formalism of quantum theory, and this classifier achieved higher performance than various classical models (Sergioli, Giuntini, & Freytes, 2019). The third paradigm, Havlíček et al. (2019) presented a quantum–classical variational classifier that is based on the concept of the variational quantum circuit. Parallel to QML evolution, various quantum DL (QDL) models have been proposed to enhance classical models (Amin, Andriyash, Rolfe, Kulchytskyy, & Melko, 2018; Beer et al., 2020; Kamruzzaman, Alhwaiti, Leider, & Tappert, 2019; Manzalini, 2019; Wiebe, Kapoor, & Svore, 2014).

Quantum subroutines are the core of quantum and QML algorithms. Quantum subroutines greatly contribute to speed up and improve the performance of algorithms, such as The Harrow–Hassidim–Lloyd (HHL) algorithm (Harrow, Hassidim, & Lloyd, 2009), for solving linear system equations with exponential speedup. The HHL algorithm is based on quantum phase estimation and quantum matrix-inversion subroutines. Grover's algorithm is based on amplitude amplification (AA) subroutine. Other subroutines include sampling, quantum annealing, and quantum Fourier transform. To implement a classical algorithm on a quantum computer, the quantum algorithm generally consists of three main phases: encoding, quantum computation, and decoding (as illustrated in Fig. 2). First, the encoding phase involves mapping data from classical to quantum states. In the second phase, quantum computation depends on the type of QML algorithms. The third step, the decoding phase involves mapping output data from quantum states to classical.

The objectives of this review are to present the most recent research on QML techniques. Besides, this review aims to analyze techniques and demonstrate the most commonly used methods in the classification task. Next, the latest QDL models have been presented. Finally, this review presents some of the challenges and future directions of QML to pave the way for researchers in this field. To sum up, we can be outlined the main contributions of this study as follows:

- Why is there progress towards QML?
- How does the concept of QC enhance classical ML?
- How to map classical data to a quantum form?
- What are different types of quantum subroutines?
- What are the future directions of QML?

The remainder of this paper is structured as follows. Section 2 presents an overview of QC, while 3 presents recent approaches in QML. QML algorithms are classified into three categories: purely QML, hybrid classical–quantum ML, and quantum-inspired ML. Section 4 summarizes QDL techniques, while Section 5 discusses quantum classification and methods of encoding data. Section 6 presents a discussion of the review questions, while Section 7 describes challenges and future directions of QML. Finally, the paper is concluded in Section 8.

2. Quantum computing

This section introduces QC concepts, such as the quantum bit, quantum gates, quantum measurement, quantum models, quantum algorithms, and quantum libraries.

2.1. Quantum bit and superposition

QC is based on postulates and characteristics of quantum mechanics (QM) (e.g., quantum bits, interference, superposition, and entanglement) for information processing. A quantum bit (qubit) can be one state, zero states, or a combination of two states at the same time known as linear superposition, unlike a classical bit, which can be represented by only one value (either 0 or 1) to process information. A qubit state is a unit vector in Hilbert space. Mathematically, the qubit state can be represented using the Bra–Ket notation. A qubit in state 0 is $|0\rangle = \begin{bmatrix}1 & 0\end{bmatrix}^{\dagger}$, while a qubit in state 1 is $|1\rangle = \begin{bmatrix}0 & 1\end{bmatrix}^{\dagger}$. A qubit is represented as a linear superposition of both basis states simultaneously:

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

where coefficients α and δ are probability amplitudes that may be complex numbers, and $|\alpha^2| + |\delta^2| = 1$.

Generally, superposition of a collection of states $|\psi_1\rangle, \dots, |\psi_n\rangle$ is as follows:

$$|\psi\rangle = \sum_{i=1}^{n} \lambda_i |\psi_i\rangle \tag{2}$$

where the complex coefficients λ_i are probability amplitudes.

2.2. Quantum gates

As per the QM postulates, the unitary operators are used to transform the state of closed quantum system state. The unitary operator maps the quantum state $|\psi\rangle$ into the state $U|\varphi\rangle$ as follows:

$$|\varphi\rangle = U\,|\psi\rangle \tag{3}$$

An operator U is a unitary transformation if the following condition is satisfied:

$$UU^{\dagger} = U^{\dagger}U = I \tag{4}$$

where $U^{\dagger}$ is the conjugate transpose of the unitary operator (U), and I is an identity operator. Quantum gates are implemented via unitary operators. The unitary operators/quantum gates act on 1−qubit, 2−qubits, or n−qubits. The quantum gate (U) with 1-qubit transforms the input state from $|\psi\rangle$ into the output state $U|\psi\rangle$. For example Not quantum gate (X-gate) transforms the input qubit from state $|0\rangle \longrightarrow |1\rangle$ state and transforms the state $|1\rangle \longrightarrow |0\rangle$ state as follows:

$$X|0\rangle = |1\rangle \tag{5}$$

$$X|1\rangle = |0\rangle \tag{6}$$

To sum up, the quantum gates can be categorized according to qubit numbers: single-qubit gates (e.g., Pauli gates, Hadamard gate), two-qubits gates (e.g., controlled-NOT (CNOT) gate, swap gate), and multi-qubit gates (e.g., Toffoli gate) (Barenco et al., 1995; Li, Meng, Zhang and Yu, 2020). Table 1 presents quantum logic gates based on the qubit number.

2.3. Quantum measurement

Quantum measurement is the essential relationship between closed quantum devices and the classical world. Projective measurement is a standard measurement in QC which can be described by projector P on quantum states in Hilbert space ($\mathcal{H}$) (Rebufello et al., 2021). Also, generalized measurement (positive operator-valued measures (POVMs)) is another type of quantum measurement (Roncaglia, Cerisola, & Paz, 2014). The quantum system changes its state after measurement. Measuring of qubit state destroys a quantum superposition.

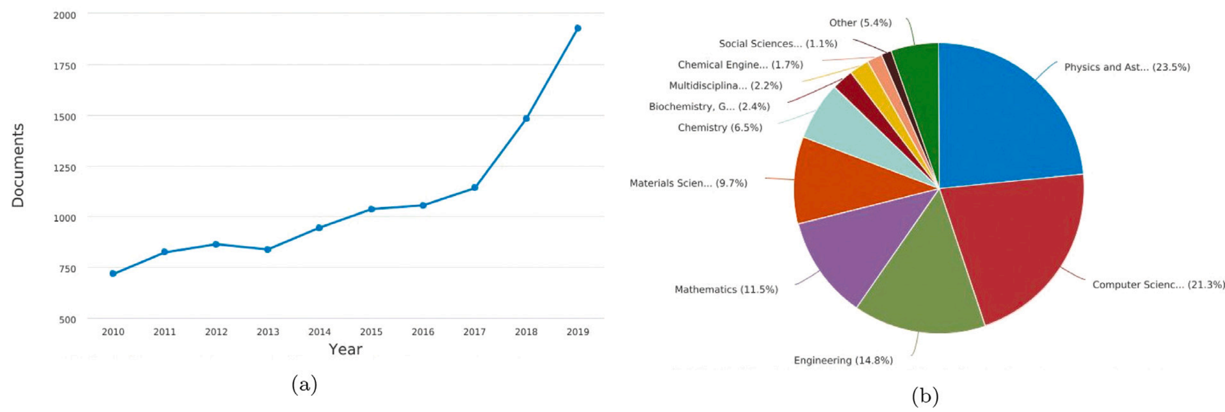


Fig. 1. Quantum computing studies published in the last 10 years [2010–2020] according to the Scopus database. 1a Number of published documents in quantum computing over the last 10 years. 1b Different areas of quantum computing.

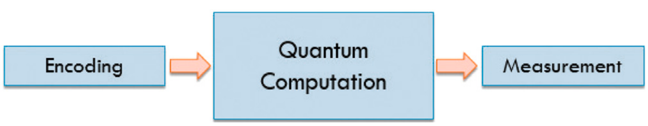


Fig. 2. The general architecture of quantum machine learning algorithms.

Based on the postulate of Von Neumann measurement of QC, the concept of measurement can be interpreted as:

$$|\psi\rangle = \sum_i \alpha_i \, |\psi_i\rangle \tag{7}$$

where $|\psi\rangle$ is the input quantum state and $\{|\psi_i\rangle\}$ is orthonormal basis of system $\mathcal{H}_A$. This quantum state will yield output from system as label i with probability $|\alpha_i|^2$.

In more general, quantum measurement can be represented by a sequence of measurement operators ($\mathrm{M_m}$) operating on quantum system state ($|\psi\rangle$) as follows (Masanes, Galley, & Müller, 2019):

$$p(m) = \left\langle \psi \left| M_m^{\dagger} M_m \right| \psi \right\rangle \tag{8}$$

where P is the probability of measurement result, and m is the possible outcomes of the measurement. After measurement, the quantum system state becomes as follows:

$$|\varphi\rangle = \frac{M_m|\psi\rangle}{\sqrt{p(m)}} \tag{9}$$

where $\sum p(m) = 1$

2.4. Quantum computing models

Because of the unique characteristics of QC, quantum models are proposed such as: adiabatic model (Albash & Lidar, 2018), quantum annealing model (Laumann, Moessner, Scardicchio, & Sondhi, 2015), circuit model (Chiribella, D'Ariano, & Perinotti, 2008; DiVincenzo, Bacon, Kempe, Burkard, & Whaley, 2000), topological model (Freedman, Kitaev, Larsen, & Wang, 2003; Lahtinen & Pachos, 2017), one-way quantum computing (Raussendorf, Browne, & Briegel, 2003; Raussendorf & Harrington, 2007), and Zidan's model (Zidan, 2020). The adiabatic model is quantum computation to solve optimization problems based on the adiabatic theorem. The quantum annealing model is used for solving optimization problems by searching over space and determining the optimal solution. The quantum circuit is a quantum computation model which is based on quantum gates (unitary transformation). A quantum circuit is a network of quantum gates followed by a quantum measurement element and linked by wires where these gates perform some unitary operators on 1−qubit or n−qubits (wires). The circuit model is the most general model for applications and solving problems; however, it is the most difficult to build. The topological model uses the features of pairs of anyons to produce unitary transformation for quantum gates. The one-way quantum model uses the high entanglement and then local measurement to destroy the entangled qubits. Zidan's model is one of the quantum computation models based on measuring entanglement characteristics that solves some of the problems (e.g., enhance speedup, and classification). QC can be used in many applications, such as cryptography, artificial intelligence, communication, optimization, ML, sampling, searching, quantum chemistry, quantum dynamics, high-energy physics, quantum internet, and metamaterials.

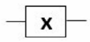
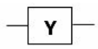
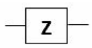
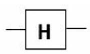
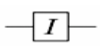
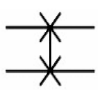
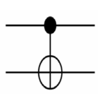
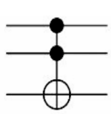
2.5. Quantum algorithms

Quantum algorithms use the postulates of QM to carry out tasks or perform a calculation. Examples include Shor's algorithm (Shor, 1999), Grover's algorithm (Grover, 1996), and Deutsch–Jozsa algorithm (Deutsch & Jozsa, 1992), Simon's algorithm (Simon, 1997), and Bernstein–Vazirani algorithm (Bernstein & Vazirani, 1997). Shor's algorithm is applied for solving factorization problems with exponential speedup, while Grover's algorithm is used to solve problems in unstructured data with quadratic speedup. The HHL algorithm is used to solve linear system equations, as illustrated in Table 2. Montanaro (2016) presented an overview of quantum algorithms and some applications in different areas.

2.6. Quantum development kit and languages

Here, we present a brief introduction to quantum programming languages and full-stack libraries. For an in-depth review, we refer the reader to Fingerhuth, Babej, and Wittek (2018) and Ying, Feng, Duan, Li, and Yu (2012). Technology companies, such as Rigetti, IBM, Q-Wave, Xanadu, Google, and Microsoft, have developed quantum software platforms and quantum languages (Hidary, 2019; LaRose, 2019). Quantum Computation Language (QCL) was the first imperative quantum programming language. The syntax of QCL is similar to that of the C programming language. Quipper (Green, Lumsdaine, Ross, Selinger, & Valiron, 2013) is a functional and high-level circuit quantum programming language for implementing quantum algorithms. The syntax of Quipper supports both procedural and declarative programming. Q language (Bettelli, Calarco, & Serafini, 2003) is the second imperative quantum programming developed by Bettelli, it is based on C++ language which has the features of an object-oriented programming paradigm. Q language has classes for basic quantum computing operations. ProjectQ (Steiger, Häner, & Troyer, 2018) is an open-source full-stack framework developed by ETH Zurich. ProjectQ library to implement quantum computing programs and algorithms by using classical or real quantum computers. Rigetti developed quantum software

Table 1
Quantum computing gates based on qubit category.

Name	Symbol	Matrix Representation	Diagonal Representation	Desc.
		Single Qubit		
Pauli-X		$\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$	$\|0\rangle\langle 1\| + \|1\rangle\langle 0\|$	Change quantum bit from state to another (Not gate).
Pauli-Y		$\begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}$	$-\underline{i}(\|0\rangle\langle 1\| - \|1\rangle\langle 0\|)$	Make π-rotation for quantum bit around the Y-axis.
Pauli-Z		$\begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$	$\|0\rangle\langle 0\| - \|1\rangle\langle 1$	Make π-rotation for quantum bit around the Z-axis.
Hadamard		$\frac{1}{\sqrt{2}}\begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$	$\frac{1}{\sqrt{2}}(\|0\rangle\langle 0\|+\|0\rangle\langle 1\|+\|1\rangle\langle 0\|-\|1\rangle\langle 1\|)$	Create superposition
Identity		$\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$	$\|0\rangle\langle 0\| + \|1\rangle\langle 1\|$	Mapping each state to itself.
		Two Qubits		
Swap		$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$	-	Swap two quantum bits states.
CNOT		$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$	-	Flips the target qubit if the control qubit is $\|1\rangle$
		Multiple Qubits		
Toffoli		$\begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}$	-	Flip target quantum bit if both control two quantum bits equal one.

Programming Toolkit called Forest to implement quantum algorithms on real quantum computers or quantum virtual machine (the analog computer) which includes pyquil library using python based on instruction quantum language (quil) (Smith, Curtis, & Zeng, 2016). IBM developed Qiskit Library (Aleksandrowicz et al., 2019; Cross, 2018), which is an open-source software development kit for quantum programs in Python. Qiskit can be divided into four modules. Cirq (Hancock, Garcia, Shedenhelm, Cowen, & Carey, 2018) is an open-source Python software library developed by Google. Microsoft developed a quantum development kit library and imperative quantum programming language called Q Sharp (Q#) (Svore et al., 2018). Strawberry Fields is a full-stack quantum software built-in Python for optimization and QML of continuous-variable circuits (Killoran et al., 2019). PennyLane (Bergholm et al., 2018) was developed by Xanadu for hybrid quantum–classical ML and optimization algorithms. Furthermore, D-Wave Systems (D-Wave, 2018) introduced an open-source platform for quantum–classical approaches, called, D-Wave Hybrid.IONQ (Grzesiak et al., 2020) is a quantum device to solve complicated real-world problems and lately, IONQ is integrated with the Qiskit IBM platform. Table 3 presents a summary of quantum technology companies and quantum libraries.

Table 2
Problems solved by quantum computing.

Problem	Solved by	Speed up
Factorization	Shor's algorithm	Exponential
Searching in unstructured data	Grover's algorithm	Quadratic
Linear systems equations	HHL algorithm	Exponential

3. Quantum machine learning

This section presents an exhaustive review of QML. Recent years have witnessed a remarkable evolution in ML-based QC. Fig. 3a presents statistics on QML from 2011 to 2020. Besides, Fig. 3b presents the distribution of QML in different areas according to the Scopus database. Many research areas are published with QML, including quantum autoencoders (Khoshaman et al., 2018; Pepper, Tischler, & Pryde, 2019; Romero, Olson, & Aspuru-Guzik, 2017), quantum biomimetics (Alvarez-Rodriguez, Sanz, Lamata, & Solano, 2018; Lamata, 2020), quantum communication (Nawaz, Sharma, Wyne, Patwary, & Asaduzzaman, 2019; Sheng & Zhou, 2017; Wallnöfer, Melnikov, Dür, & Briegel, 2020), quantum annealing (Li, Di Felice, Rohs,

Table 3
Summary of quantum technology companies and quantum libraries.

Company	Library	Supported operating system	URL
Rigetti	PyQuil and Grove		https://www.rigetti.com
IBM	Qiskit		https://quantum-computing.ibm.com
Xanadu	Strawberry fields and PennyLane		https://pennylane.ai
Google	Cirq	Windows, Linux, and Mac	https://cirq.readthedocs.io/
D-Wave	D wave Hybrid		https://www.dwavesys.com/ https://cloud.dwavesys.com/leap/login/
Microsoft	Quantum Development Kit (QDK)		https://docs.microsoft.com/en-us/quantum/
ETH Zurich	ProjectQ		https://projectq.ch/
IonQ	–	–	https://ionq.com/

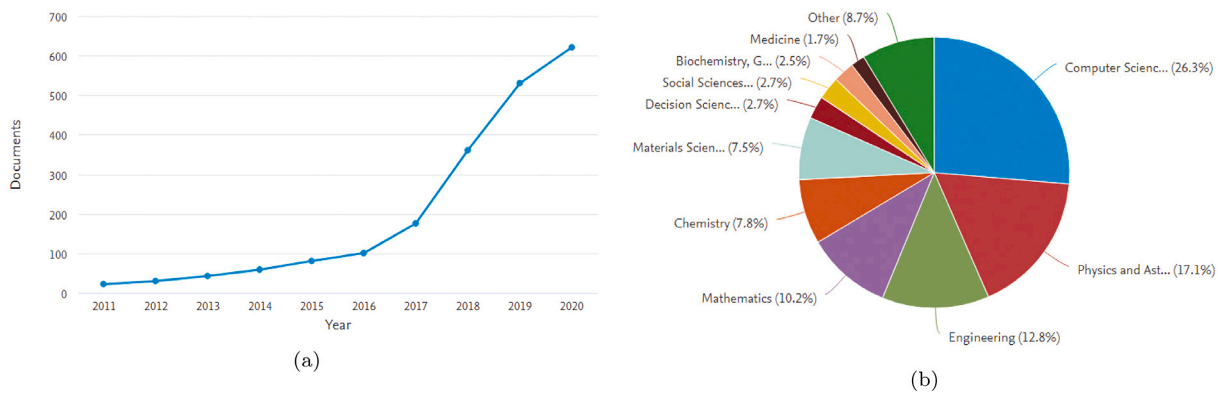


Fig. 3. Quantum machine learning studies published in the last 9 years [2011–2020] according to the Scopus database. 3a Number of published documents in quantum machine learning over the last 9 years. 3b Different areas of quantum machine learning.

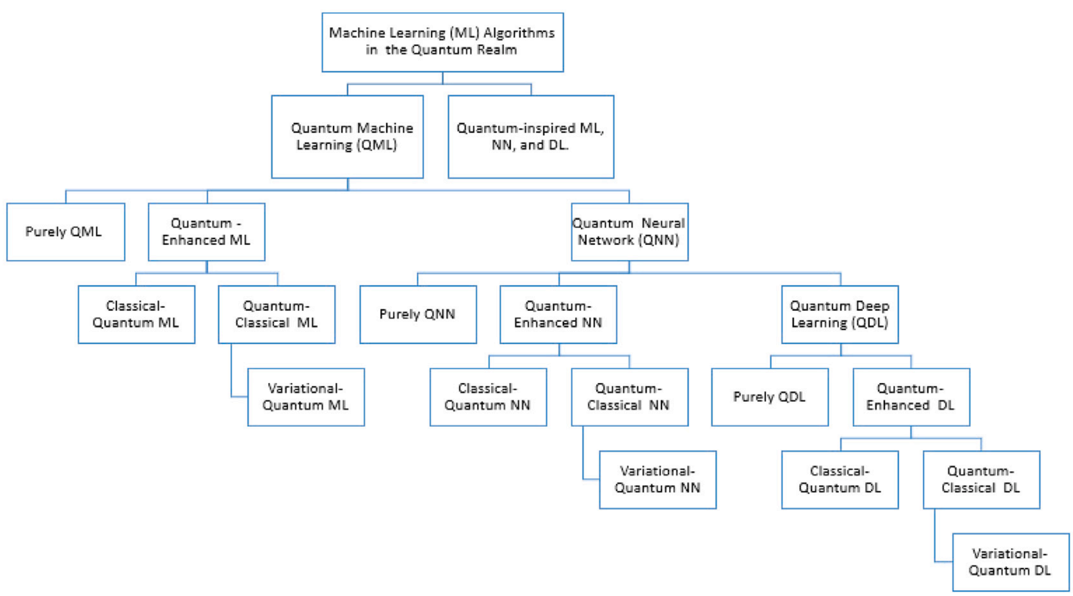


Fig. 4. Taxonomy of machine learning algorithms in the quantum domain.

& Lidar, 2018; Rieffel et al., 2015), computational chemistry (McArdle, Endo, Aspuru-Guzik, Benjamin, & Yuan, 2020; Von Lilienfeld, 2018), and Boltzmann machine (Amin et al., 2018). In this study, QML algorithms can be organized and classified into three categories according to the used idea of quantum computing with ML. In this review, we focus on three approaches of ML in the quantum domain: Purely QML (Beer et al., 2020; Dunjko, Taylor, & Briegel, 2016; Levine, Sharir, Cohen, & Shashua, 2019), hybrid classical–quantum ML (Killoran et al., 2019; Mari, Bromley, Izaac, Schuld, & Killoran, 2020), quantum-inspired ML (Gao, Ma, Song, & Liu, 2017; Pomarico et al., 2021). Table 4 summarizes the existing QML algorithms.

Various review papers have presented the quantum version of ML algorithms as shown in Fig. 4. For example, Jeswal and Chakraverty (2019) presented various quantum neural networks (QNNs) techniques and their applications in real-world problems. Also, the authors reported that QNNs are more powerful than classical neural networks and can enhance computational efficiency. In another review, Benedetti, Lloyd, Sack and Fiorentini (2019) provided an overview of hybrid quantum–classical models based on parameterized quantum circuits. Also, the review presented the application of hybrid systems in supervised and generative learning. The review also provided a framework for components of models (e.g., variational circuit, encoder circuit). Ciliberto et al. (2018) presented a review of QML as well as challenges. Besides, the authors provided some quantum subroutines, especially quantum linear algebra, and demonstrated how a quantum computer works with data.

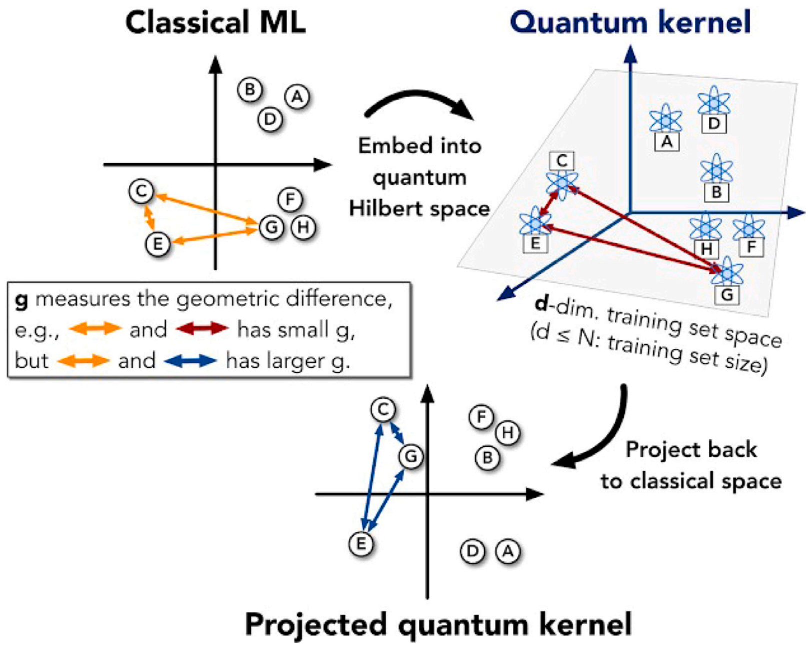


Fig. 5. The projected quantum kernel proposed by Huang et al. (2021).

In this study, we introduce the recent works of the three QML approaches. Besides, this review presents state-of-the-art works in QDL. Moreover, this study provides different quantum subroutines. Finally, this review presents many future directions of QML (see Fig. 4).

3.1. Purely quantum machine learning

In Huang et al. (2021) the authors used the potential quantum advantage to propose a new approach in ML tasks. This approach is based on the geometric kernel function via the input data space. Also, they provided a quantum kernel in classical space using quantum and classical ML models called ''projected quantum kernels'' (PQK). This quantum kernel measures the similarity between data and gives strict quantum speed-up in learning tasks. Fig. 5 shows the proposed projected quantum kernel. The geometric constant (g) measures the geometric difference in classical and quantum ML algorithms with various kernel functions. They reported that the potential quantum advantage is based on the amount of data.

In Schuld, Sinayskiy, and Petruccione (2016), the authors proposed a new quantum algorithm for pattern recognition based on supervised learning. This algorithm is a version of linear regression called, quantum linear regression. The authors used the amplitude encoding method to convert data in the quantum state. The quantum linear regression works on quantum data with N-dimensions of features in logarithmic time. Rebentrost et al. (2014) presented a quantum version of the support vector machine (QSVM) for the classification of big data. The quantum SVM is based on a non-sparse matrix to carry out matrix inversion of inner product for training big data. It works with a large number of features and samples with logarithmic complexity. The Quantum SVM of big data achieves exponential speedup on the feature space dimensions and samples number compared to classical SVM.

Recently, the NNs are presented in QC perspective (Altaisky, 2001; Gupta & Zia, 2001; Schuld, Sinayskiy, & Petruccione, 2014). The QNNs advantages over classical NNs are discussed in Ezhov and Ventura (2000) (e.g., quantum parallelism, higher stability, higher information processing speed, and memory capacity). da Silva, Ludermir, and de Oliveira (2016) introduced a new QNNs called a quantum perceptron over a field (QPF) and its learning algorithm, Superposition-based Architecture Learning (SAL). The SAL algorithm is based on a superposition feature and quantum operator. Besides, it processes NN architecture with polynomial time. QPF overcomes the limitations of quantum perceptron models. In another work, the authors (Schuld, Sinayskiy, & Petruccione, 2015b) introduced a quantum version of classical perceptron using quantum phase estimation on quantum hardware. The quantum perceptron algorithm simulates the activation function (step function) in NNs.

In competitive NNs, Zhou (2010) presented two main parts: first, a new model based on competitive learning NNs with QC called QCNN. The QCNN model classifies input patterns using quantum pattern competition. In the second part, Zhou provided the capacity of memory for the proposed QCNN. The QCNN achieves competitive learning using a quantum register without network weight. Another QCNN model using quantum entanglement and Grover's algorithm is proposed (Zhong & Yuan, 2012). This model used quantum associative memory due to the pseudo patterns. Also, this model recalls the pseudo-states during the competition process in incomplete patterns. The authors in Zidan et al. (2019) proposed another QCNN based on entanglement measure for binary classification called QCPNN. The QCPNN classifies input data in incomplete patterns on a quantum computer. The entanglement measure is used to handle the competition between neurons using wining-take-all to determine the winning class. The QCPNN overcomes the disadvantage of the QCNN model (Zhong & Yuan, 2012) (recall pseudo-states).

3.2. Hybrid classical–quantum machine learning

Lately, The authors in Abbas et al. (2021) discussed the power of QNN with the current near-term quantum hardware. The authors proposed a new measure for the capacity of the model they named, the effective dimension. This effective dimension is used to bound the ability of the model to generalize on new/unseen data. In addition, they reported their measure is a data-dependent generalization method with a fisher information matrix. Finally, the authors reported that the QNN achieved faster training with the current noisy quantum device compared to the classical NN. They also showed QNN is more capable than classical NN. Chen and Yoo (2021) proposed a new training model that relies on federated learning and hybrid quantum–classical ML, called, federated QML. The authors used the feature of quantum devices to solve the rising privacy problem with limited available quantum hardware. The authors used the quantum hardware (i.e., read device or simulator) as local clients. Also, the authors utilized classical–quantum transfer learning with VGG16 for feature extraction. The advantage of the proposed framework works on classical and quantum data.

Willsch, Willsch, De Raedt, and Michielsen (2020) implemented an SVM on a quantum annealer device (DW2000Q), called, QA-SVM. The authors used the quantum annealer to train and optimize the SVM based on the "Quadratic unconstrained binary optimization" (QUBO) equation to minimize cost energy. Moreover, the authors utilized some quantum annealing features (e.g., reverse annealing and special annealing schedules) to improve the final results. In Chakraborty, Shaikh, Chakrabarti, and Ghosh (2020), the authors introduced new quantum algorithms based on many subroutines as a quantum oracle, counting, AA, and quantum amplitude estimation for feature selection (HQFSA) to enhance the performance of ML techniques. The proposed algorithm achieved quadratic time complexity and better performance in some cases. The main disadvantage of HQFSA is that it runs on a quantum simulator only.

Dang, Jiang, Hu, Ji, and Zhang (2018) proposed a new quantum model for image classification, called, quantum KNN algorithm. The quantum KNN model consists of two parts: classical and quantum part. The authors used the classical computer to extract the features of images. The extracted features are converted into a quantum state by a quantum device. Then, the quantum circuit is used to calculate the similarity between images. Finally, the classification process is performed by a measurement circuit. The quantum KNN model outperforms classical models in terms of efficiency and classification performance. Adhikary, Dangwal, and Bhowmik (2020) used a quantum circuit to present a new variational quantum classifier with a single quantum system and encode N-dimensional data with a training algorithm called single-shot training. Also, the authors encoded the whole dataset into a single quantum state. The single-shot training uses fewer parameters for training and achieves higher precision. Mitarai, Negoro, Kitagawa, and Fujii (2018) presented a hybrid classical–quantum technique to perform different tasks such as classification, regression, and clustering called, quantum circuit learning (QCL). The QCL acts on small-scale quantum devices. The authors reported The ability of QCL to perform with high-dimensional classification/regression tasks and carry out quantum many-body dynamics as well.

Havlíček et al. (2019) proposed two quantum techniques in supervised ML called, variational quantum classifier (VQC) and quantum kernel estimator (QKE). The quantum state space is used as feature space in VQC and QKE. The VQC is based on a quantum variation circuit which presented in Farhi and Neven (2018) and Mitarai et al. (2018) for data classification in a way like classical SVMs. The VQC consists of two stages (training and classification stage). In the training stage, the authors used four steps to compute the hyperplane between training data. The classification stage is used to classify new data with the correct label. The second technique, the main idea of QKE is to estimate a quantum kernel on quantum device and using this kernel with classical SVMs. The quantum kernel function is hard to estimate with classical devices. Schuld and Killoran (2019) proposed two-hybrid quantum techniques for classification problems. They demonstrated that QC enhances classical ML algorithms, such as kernel methods. QC performs complex computations in Hilbert space more efficiently. The authors focused on using feature maps and kernel methods in the field of QC.

3.3. Quantum-inspired machine learning

Tiwari and Melucci (2019) introduced a new quantum-inspired binary classifier, whose fundamental idea is based on decision theory, classical ML, and the theory of quantum detection that utilizes one of the laws of QM (i.e., superposition) to increase the degree of freedom in decision-making. The proposed classifier achieved high precision, recall, and F-score comparable with k-nearest neighbors (KNN), SVM, and other classical techniques. Sergioli et al. (2019) proposed a novel quantum-inspired classifier for binary supervised learning called Helstrom Quantum Centroid based on density matrices and formalism of quantum information theory. The authors evaluated the performance of their model on 14 datasets compared to different classical models.

Sergioli et al. (2018) introduced the quantum nearest mean classifier based on the principle of the classical minimum distance classifier. The algorithm consists of three main steps. The first is density pattern (encoding) to transform each classical data point into a quantum object. The second step is the quantum centroid to calculate the distance between density patterns to classify unknown quantum objects into the correct class. Lastly, the third step is decoding, which aims to transform the final classification result into the classified data. The algorithm achieved higher accuracy in many medical datasets than the classical counterpart on a cancer dataset.

Lu and Braunstein (2014) proposed a quantum version of a decision tree classifier. The quantum model is based on quantum entropy impurity and the quantum fidelity measure. The authors utilized von Neumann entropy for the splitting process. Also, the authors provided a clustering technique in the quantum domain to discretize the training data in the quantum state. The authors introduced the open issues, which related to decision tree algorithm with quantum computing. In Sagheer, Zidan, and Abdelsamea (2019), the authors introduced a novel quantum-inspired NN called, the autonomous perceptron model (APM). The APM depends on the quantum bit feature. The proposed model achieved higher accuracy with lower time complexity than other traditional algorithms. Ruan, Xue, Liu, Tan, and Li (2017) presented a quantum KNN algorithm based on the Hamming distance matrix. Quantum KNN is a version of the KNN algorithm to avoid the defects of the simplifying assumption for classification problems. QKNN outperformed Centroid and QNN in terms of performance and classification accuracy.

As briefly, Table 5 summarizes quantum classification algorithms in terms of the classification type, basic principle of the algorithm, quantum or classical data, data-sets, and advantages.

4. Quantum deep learning

In this section, we present recent publications on QDL. In the last several years, remarkable research has occurred in DL-based QC. Fig. 6a presents the statistics on QDL from 2011 to 2020. Besides, Fig. 6b presents the distribution of QDL in different areas according to the Scopus database. To the best of our knowledge, the first concept of QDL appeared with reinforcement learning inspired by superposition principle (Dong, Chen, & Chen, 2005; Dong, Chen, Li, & Tarn, 2008).

In Li, Zhou, Xu, Luo and Hu (2020), the authors proposed a hybrid quantum–classical deep CNN based on the quantum variational circuit, called, QDCNN. The structure of the QDCNN is similar to that of a CNN. QDCNN consists of convolutional and classification parts. In other work, the authors of Cong, Choi, and Lukin (2019) introduced a new quantum CNN (QCNN) inspired by the layers of a classical CNN using a quantum circuit. The key principle of the QCNN algorithm is based on the quantum circuit and unitary transformations (quantum gates). Similar to the architecture of CNN, a QCNN consists of a convolutional layer, pooling layer, and fully connected layer. The authors used quasi-local unitaries (U_i) as the convolutional layer. The pooling layer is designed as unitaries (V_j) to reduce the output features of unitaries (U_i). The unitary (F) is used as a fully connected layer. Fig. 7 presents the architecture of the QCNN.

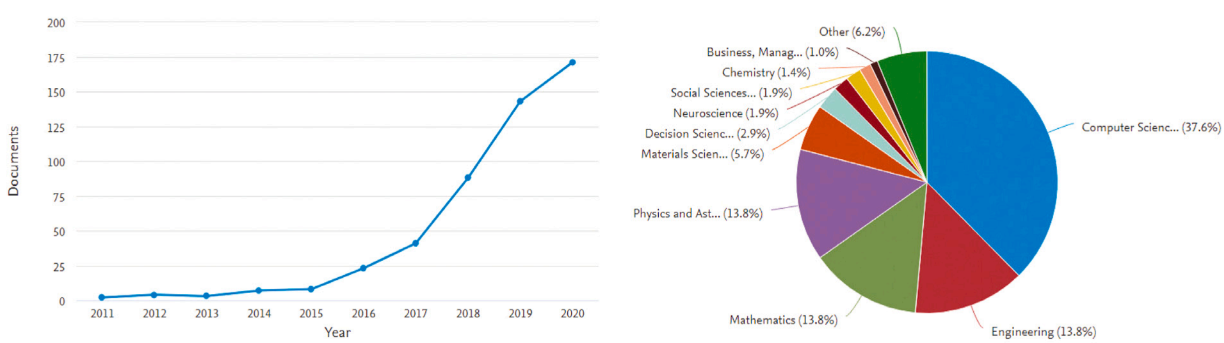
In another hybrid study, The authors of Henderson, Shakya, Pradhan, and Cook (2020) used quantum circuits with standard convolutional neural networks for image classification. The authors used a small-depth quantum circuit to implement on available small-scale and NISQ quantum hardware. The quantum circuit is applied as a convolution layer to extract informative features. There are three stages in the quantum convolutional layer: encoding, quantum circuit, and

Table 4
Existing quantum machine learning techniques.

Ref.	Year	Model	Class	Task	Application
Huang et al. (2021)	2021	PQK	Purely	Learning approach	Fashion-MNIST
Chen and Yoo (2021)	2021	Federated QML	Classical–quantum	Training model	Cats vs. Dogs and CIFAR dataset
Willsch et al. (2020)	2020	QA-SVM	Classical–quantum	Classification and optimization	Real and synthetic data
Rebentrost et al. (2014)	2014	QSVM	Purely	Classification	Big data
da Silva et al. (2016)	2016	QPF	Purely	Classification	–
Schuld et al. (2016)	2016	Quantum linear regression	Classical–quantum	Regression	–
Zidan et al. (2019)	2019	QCPNN	Purely	Classification	Risky Nuclear Power Plant
Tiwari and Melucci (2019)	2019	QIBC	In-spired	Classification	Text and image Corpora data set
Sergioli et al. (2019)	2019	HQC	In-spired	Classification	–
Sergioli et al. (2018)	2018	QNMC	In-spired	Classification	Medical Data
Dang et al. (2018)	2018	QKNN	In-spired	Classification	Graz-01 and Caltech-10
Lu and Braunstein (2014)	2014	Quantum decision tree	In-spired	Classification	–
Sagheer et al. (2019)	2019	APM	In-spired	Classification	Breast cancer and synthetic data
Adhikary et al. (2020)	2020	Single-shot training	Classical–quantum	Classification	Cancer, Sonar, and Iris
Chakraborty et al. (2020)	2020	HQFSA	Classical–quantum	Feature selection	Breast Cancer, Iris, Wine, Vehicle, Glass, Sonar, and Ionosphere
Havlíček et al. (2019)	2019	QVC	Classical–quantum	Classification	–
Havlíček et al. (2019)	2019	Quantum-kernel SVM	Quantum	Classification	–
Schuld and Killoran (2019)	2019	–	Classical–quantum	Classification	–
Ruan et al. (2017)	2017	QKNN	Classical–quantum	Classification	MNIST digit
Zhong and Yuan (2012)	2012	QCNN	Purely	Pattern recognition	
Zhou (2010)	2010	QCNN	Purely	Pattern recognition	–

Table 5
Comparison between quantum machine learning techniques.

Algorithm	Classical type	Based idea	Quantum data	Data-sets	Advantages	Limitations
QIBC	Binary	Quantum detection theory	√	Image and text corpora	High precision, recall, and F1-measure. Less computation time compared to SVM and KNN.	Need to many features to improve performance. Increase computation cost with large number copies.
HQC	Binary	Distinguish the ability between quantum states	√	Appendicitis and other data sets	Improve classification rate.	Do not work efficiently with many features vectors in large data.
QNMC	Binary and multi-class	Quantum formalism	√	Diabetes, Liver Cancer and Ionosphere	Achieve higher accuracy than classical NMC.	Do not achieve better performance with cancer data.
VQC	Binary and multi-class	Quantum Variational circuit	√	Breast Cancer and wine data	Higher performance, Classify binary, and multi-class efficiently.	High cost
QSVM	Binary and multi-class	Run on real quantum device	√	Breast Cancer and wine	Better accuracy from classical SVM.	Difficult implementation.
APM	Binary and multi-class	Quantum Bit	X	Breast Cancer, Wine Vintage, and synthetic data	Classify specific non linear problems with one neuron only. Take a few instances to train model and achieve higher accuracy.	Work on classical computer only.



(a)   (b)

Fig. 6. Quantum deep learning studies published in the last 9 years [2011–2020] according to the Scopus database. 6a Number of published documents in quantum deep learning over the last 9 years. 6b Different areas of quantum deep learning.

measurement. In another architecture, Bausch (2020) proposed a quantum version of the recurrent neural network (RNN) called QRNN. The essential component of QRNN is a quantum neuron. The QRNN is used to classify digit data. Besides, QRNN is used as a generative model.

Benedetti et al. (2019) introduced a framework generative model called data-driven quantum circuit learning (DDQCL). The DDQCL approach is an unsupervised hybrid quantum–classical approach for the characterization of noisy intermediate-scale quantum (NISQ) hardware for solving sampling problems, such as bars and stripes and random thermal datasets using shallow quantum circuits. Using the quantum computer, the authors (Zhao, Pozas-Kerstjens, Rebentrost, & Wittek, 2019) proposed a new version of the Bayesian technique for deep

Table 6
Existing quantum deep learning models.

Ref.	Year	Model	Class	Task	Classical Arch.	Application
Li, Zhou et al. (2020)	2020	QDCNN	Classical–quantum	Image recognition	CNN	Mnist and GTSRB
Cong et al. (2019)	2019	QCNN	Purely	Classification	CNN	Quantum phase recognition and quantum error correction
Henderson et al. (2020)	2020	HQCNN	Classical–quantum	Classification	CNN	–
Bausch (2020)	2020	QRNN	Purely	Digit data classification	RNN	MNIST data
Benedetti, Garcia-Pintos et al. (2019)	2019	DDQCL	Classical–quantum	Generative model	DL	Synthetic data

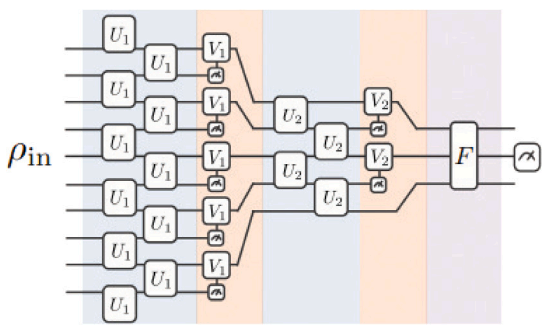


Fig. 7. The quantum convolutional neural networks architecture proposed by Cong et al. (2019).

learning. The main part of this technique is quantum matrix inversion. This technique is implemented on two quantum hardware (Rigetti and IBM). To sum up, Table 6 summarizes QDL techniques.

5. Quantum classification

In this section, the concept of quantum classification and encoding methods are introduced.

5.1. Classification in quantum world

Classification is a popular task in the supervised learning domain that maps given inputs data (x) to a discrete target output (y) through a function approximation f as follows: $y = f(x)$. In general, the main goal of classification is an accurate prediction of class via a discrimination function. Classification problems can be classified into two main problems. Classification with two classes is known as binary classification (e.g., cancer diagnosis, spam detection), whereas classification with more than two classes is known as multi-class classification (e.g., image and digit classification). In supervised learning, classification problem can be represented as: target labels $C = \{y_1, y_2, \ldots, y_N\}$ and data in training phase can be described as $D_N = \{(x_1, y_1), \ldots, (x_i, y_i), \ldots, (x_N, y_N)\}$ where x_i is a vector of features of ith instance (i), N is number of features, and y_i is the target label corresponding to the instance x_i, where $y_i \in C$. In the case of binary classification $y_i \in \{y_1, y_2\}$ and $x_i \in R^d$. In the case of multi-class classification $y_i \in \{y_1, \ldots, y_N\}$ where $x_i \in R^d$ and d is real-valued attributes.

To describe classification problem in the QML domain, classical data should be converted into quantum data and can then be represented in the training data as $D_N = \{(|\psi_1\rangle, y_1), \ldots, (|\psi_i\rangle, y_i), \ldots, (|\psi_N\rangle, y_N)\}$ where $|\psi_i\rangle$ is the order (i) of the quantum state of D_N, $|\psi_i\rangle \in C^{2^d}$ and $y_i \in \{y_1, y_2\}$ (Gambs, 2008; Lu & Braunstein, 2014). In the case of binary classification, a question is how to map data to a quantum form. There is more than one method for encoding classical data into quantum data, such as basis encoding and amplitude encoding.

5.2. Encoding methods

There are various methods for encoding classical data into quantum data in a Hilbert space. Encoding data signifies loading classical data into a quantum computer (quantum states) (Mott, Job, Vlimant, Lidar, & Spiropulu, 2017; Schuld, 2018; Schuld & Killoran, 2019).

5.2.1. Basis encoding

Basis encoding is the simplest method to encode data into quantum data. This method associates each classical input with the computational basis of the quantum state. For example, the (1100) classical input string is encoded into four qubits $(|1100\rangle)$ quantum states. In general, to encode a dataset by the basis method or to represent data in the computational basis states of qubits, the following equation must be applied:

$$|D\rangle = \frac{1}{\sqrt{\mathrm{M}}} \sum_{m=1}^{M} |X^m\rangle \tag{10}$$

where $D = X^1, X^2, \ldots, X^M$ represents classical data that are in the form of a binary string, $X^m = b_1, b_2, \ldots, b_N$, $b_i \in 0, 1$, $i \in 1, 2, \ldots, N$, N is the number of features, and M is the number of samples in data-set (D).

5.2.2. Amplitude encoding

Amplitude encoding is the most commonly used and popular method of encoding in QML algorithms. The main principle of amplitude encoding is based on the association of classical data with quantum state amplitudes. To encode a classical data vector into quantum amplitudes, the classical vector must be converted into a normalized classical vector (Sergioli et al., 2019).

$$\mathrm{X} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_{2^n} \end{bmatrix} \tag{11}$$

where X is a normalized classical vector, $x \in C(2^n)$, and C is complex numbers.

$$|\psi_x\rangle = \sum_{i=1}^{2^n} x_i |\mathrm{i}\rangle \tag{12}$$

where $|\psi\rangle \in$ Hilbert space$(\mathcal{H})$ and $\sum_i |x_i|^2 = 1$.

5.2.3. Angle encoding

Classical data has been converted into rotation angles of quantum states using angle encoding. Rotation angles are rotation operators of Pauli gates around X, Y, Z axes. The RX, RY, and RZ rotation angles can be expressed as the following equations.

$$R_x(\alpha) = \begin{pmatrix} \cos\left(\frac{\alpha}{2}\right) & -i\sin\left(\frac{\alpha}{2}\right) \\ -i\sin\left(\frac{\alpha}{2}\right) & \cos\left(\frac{\alpha}{2}\right) \end{pmatrix} \tag{13}$$

$$R_y(\alpha) = \begin{pmatrix} \cos\left(\frac{\alpha}{2}\right) & -\sin\left(\frac{\alpha}{2}\right) \\ \sin\left(\frac{\alpha}{2}\right) & \cos\left(\frac{\alpha}{2}\right) \end{pmatrix} \tag{14}$$

$$R_z(\alpha) = \begin{pmatrix} e^{-i\frac{\alpha}{2}} & 0 \\ 0 & e^{i\frac{\alpha}{2}} \end{pmatrix} \tag{15}$$

6. Discussion

QML has become a new research field and has appeared in many applications. The progress and success of QC are widely observed. Thus, the advantages and properties of QM should be applied to ML. To the best of our knowledge, the first concept of QML applied in Chrisley

Algorithm \ Data	Quantum	Classical
Quantum	QQ	QC
Classical	CQ	CC

Fig. 8. Quantum machine learning algorithms based on quantum/classical data and quantum/classical algorithm.

(1995). The QC is mixed with supervised and unsupervised learning in Aïmeur, Brassard, and Gambs (2006) and Lloyd, Mohseni, and Rebentrost (2013). It is noteworthy, the authors in Aïmeur et al. (2006), Dunjko et al. (2016) and Schuld (2018) have divided QML algorithms into four categories depend on the integration of quantum or classical algorithm, and quantum or classical data as shown in Fig. 8. The four categories can be defined as follows:

- The quantum–quantum (QQ) category, this category is also known as purely QML. The QQ category uses quantum algorithms and data.
- The quantum–classical (QC) category, this category uses a quantum algorithm to learn from classical agents (Kuo, Fang, & Chen, 2021).
- The classical–quantum (CQ) category, CQ algorithms are quantum versions of standard ML, and these algorithms can be executed on a real quantum device
- The classical–classical (CC) category, commonly used by quantum-inspired ML category. The using quantum computing features (i.e., quantum bits, superposition, and entanglement) in the CC category by inspiration.

Remarkably, The QML field is the big picture, which contains quantum–quantum, classical–quantum, and quantum–classical ML.

From the above-mentioned studies in Sections 3 and 4, the used quantum advantage/principle is the major idea to classify QML algorithms. The approach name is based on the used quantum idea. quantum-inspired ML is also known as quantum-like ML. quantum-inspired and variational-quantum ML are also known as the quantum-enhanced ML approach. And notably, quantum–classical variational ML is a sub-approach of the classical–quantum ML approach. To conclude, the combination between quantum and classical machine and deep learning approaches can be established various QML approaches.

The advantages and limitations of the classical–quantum ML approach: the classical–quantum ML algorithms are implemented on a quantum computer/simulator. These algorithms act on quantum data, which needs to apply quantum encoding techniques (see encoding methods 5.2). The main limitation of this approach is quantum hardware. Indeed, The currently quantum available hardware, small-depth circuit, and limited qubits number are a big challenge task with classical–quantum algorithms (Chen & Yoo, 2021). Due to the restricted hardware, the advantages of the classical–quantum ML (Li et al., 2018) (e.g., run-time improvement, and learning from less data) are not clear enough in real-world applications (Houssein, Abohashima, Elhoseny, & Mohamed, 2021; Phillipson, Wezeman, & Chiscop, 2020; Sierra-Sosa, Arcila-Moreno, Garcia-Zapirain, Castillo-Olea, & Elmaghraby, 2020).

The advantages and limitations of quantum-inspired ML approach: The main limitation of the quantum-inspired ML algorithms is that they operate on classical computers only (Sagheer et al., 2019). The advantage of quantum-inspired ML is improved performance. From the above studies, in general, the classical–quantum approach algorithms improve in time complexity (Schuld et al., 2016; da Silva et al., 2016). While quantum-inspired algorithms improve classical ML in terms of classification accuracy (Ruan et al., 2017; Tiwari & Melucci, 2019), especially in medical applications (Sergioli et al., 2018). Also, the similarities and differences between QML and quantum-inspired algorithms have been widely discussed (Sergioli, 2020).

We now address our review questions. The answer to the first question of why is there progress towards QML is to solve challenges and difficult tasks that remain in classical ML. With the increasing size of data, time and memory consumption are also increased during the learning phase. However, achieving low-cost learning with higher accuracy and performance is a challenging task using classical algorithms. It is difficult to estimate kernel functions in higher dimensions (Hansen, 2009; Silverman, 2018; Wang et al., 2016). Besides, it is difficult to determine the eigenvectors (Wang et al., 2016) and solve complex optimization problems.

QC has demonstrated superiority in solving intractable problems (e.g., factorization and searching in an unstructured database). According to Dunjko and Briegel (2018), QML can improve run-time and efficiency, and capacity Learning. D-wave introduced quadrant algorithms based on a central processing unit, graphics processing unit, and quantum annealing computer. Quadrant ML algorithms can achieve higher performance with low-cost training data on large data compared to classical DL and traditional algorithms (Li et al., 2018).

The second review question asks how the concept of QC enhances classical ML. QC can enhance traditional algorithms using two different methods. The first method implements classical algorithms on quantum computers/simulators. This method must encode classical data into quantum data. The second method develops QML algorithms depending on quantum/subroutines algorithms, such as AA, Grover's algorithm, quantum matrix inversion, QPE, variational quantum circuit (Schuld, Bocharov, Svore, & Wiebe, 2020), quantum annealing, and sampling (Benedetti, Garcia-Pintos et al., 2019).

6.1. Quantum subroutines

Now, we demonstrate several quantum subroutines to enhance the performance of QML.

6.1.1. Quantum Fourier transform

QFT (Coppersmith, 2002) is the backbone of other quantum algorithms, such as the QPE algorithm and Shor's algorithm. QFT is a quantum version of inverse discrete Fourier transform and achieves exponential speedup. The principle of QFT depends on mapping amplitudes of the current quantum state (R) onto the next amplitudes of the quantum state (Q): $|\mathrm{R}\rangle = \sum_{i=1}^{N-1} x_i |\mathrm{i}\rangle$, $|Q\rangle = \sum_{j=1}^{N-1} y_j |j\rangle$ where x_i, y_j are complex numbers, and the mapping equation is as follows (Wang et al., 2016):

$$\mathrm{Qj} = \frac{1}{\sqrt{N}} \sum_{n=1}^{N-1} X_n e^{2\pi i \frac{jk}{N}} \tag{16}$$

Where $\mathrm{j} = 1, 2, \ldots, \mathrm{N}-1$

6.1.2. Quantum phase estimation

QPE (Cleve, Ekert, Macchiavello, & Mosca, 1998; Luis & Peřina, 1996) is one of the most important subroutines in many quantum algorithms (Shor, 1999) and QML algorithms. The QPE algorithm is based on the QFT subroutine. The goal of this algorithm is to find the eigenvalue ($e^{2\pi i\theta}$) of the unitary matrix (U) or the optimal approximation of the phase (θ) using the following equation:

$$U|\varphi\rangle = e^{2\pi i\theta}|\varphi\rangle \tag{17}$$

where $0 \le \theta < 1$ and $|\varphi\rangle$ is the eigenvector.

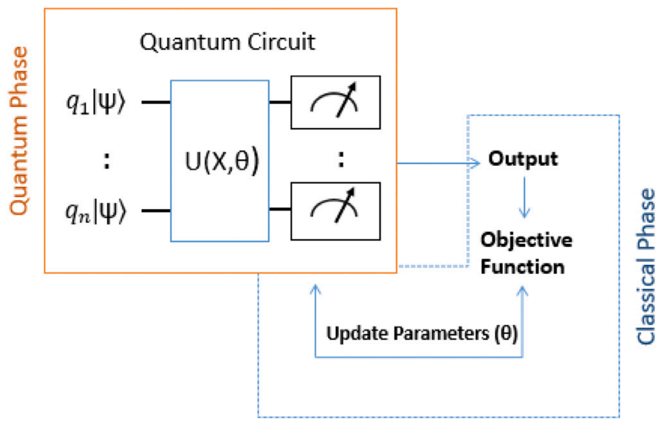


Fig. 9. Two phases of the quantum variational circuit for hybrid quantum–classical applications in quantum machine learning and quantum neural networks. The quantum circuit phase consists of state preparation, circuit, and measurement. Another phase is a classical phase that includes output after measurement and consists of a cost function and learning algorithms that update parameters (θ).

6.1.3. Amplitude amplification

AA (Brassard, Hoyer, Mosca, & Tapp, 2002) (also called quantum interference) is a fundamental subroutine and the key principle of Grover's algorithm. Grover's algorithm is a subroutine for many different algorithms. The key objective of AA is to increase the solution probability of the amplitude (P) from an arbitrary state to the target state over-all iterations (Ambainis, 2012; Brassard, Høyer, & Tapp, 1998). The success probability of the amplitude can be increased using the following formula:

$$1-\left(\frac{m^2}{3}(P)\right)\left(m^2(P)\right) \tag{18}$$

where m is the number of iterations and P is the success probability.

6.1.4. Variational quantum circuit

Variational quantum circuits, also known as parameterized quantum circuits, are used by most hybrid quantum–classical algorithms (McClean, Romero, Babbush, & Aspuru-Guzik, 2016). The central idea of a variational circuit is to optimize parameters according to an objective function. As illustrated in Fig. 9, variational quantum circuits consist of two phases: a quantum phase and a classical phase. The quantum phase includes state preparation and a quantum circuit. This circuit is the heart of the variational circuit, which parameterizes input (X) based on the number of parameters (θ) and measurement. The classical phase includes the output of the circuit, objective function, and learning algorithm. The quantum variational circuit can be optimized by classical optimization algorithms (e.g., gradient descent, stochastic gradient descent, particle swarm optimization). Some uses of the variational quantum circuit include ML (Havlíček et al., 2019; Schuld & Killoran, 2019), classification (Chen, Wei, Zhang, Yu and Yoo, 2020), optimization (Moll et al., 2018), DL (Chen et al., 2020), speech recognition (Yang et al., 2020), and metric learning (Nghiem, Chen, & Wei, 2020). Table 7 summarizes the quantum subroutines discussed in this paper and their applications. We note that QFT and the variational quantum circuit are the most commonly used subroutines in most algorithms and applications.

7. Challenges and future directions

In this section, we outline challenges and future directions in QML, such as small-scale quantum computers, limited quantum bits, encoding methods, and new QML techniques.

7.1. Small-scale quantum computer

A key challenge is to build quantum computers with a large number of qubits to implement and test QML algorithms and work with large data in the near future. Fig. 10 displays the numbers of qubits achieved by different technology companies, such as Rigetti, IBM, Q-Wave, Xanadu, Google, and Microsoft. Up to now, quantum computers have been developed on a small scale, which restricts the use of a limited amount of data (Herbster, Mountney, Piat, & Severini, 2020). Thus, researchers have developed algorithms that work on available small-scale and NISQ quantum hardware compatible with the available number of qubits (Preskill, 2018). A limited number of qubits using a smaller number of features leads to the loss of many important data. Besides, with limited qubits, big data processing cannot be applied on quantum devices (Perdomo-Ortiz, Benedetti, Realpe-Gómez, & Biswas, 2018).

7.2. Encoding methods

Encoding data into quantum states is one of the main QML challenges. This process requires high time consumption and power for mapping classical data to quantum data (e.g., images, big data). Thus, designing new techniques for encoding data is an interesting future research direction. In 2020, LaRose and Coyle (2020) presented a robust binary quantum classifier with noisy quantum states based on the selection of the best encoding method to load data into the quantum system. Furthermore, the authors discussed different methods for encoding with quantum binary classifiers. The authors also applied several encoding methods on the same data and demonstrated that encoding data techniques can improve model accuracy.

7.3. Machine learning based quantum subroutines/algorithms

In the near future, researchers will develop new theoretical and applied ML algorithms compatible with quantum hardware. The using quantum information theory or quantum subroutines aims to solve ML problems and improve performance. Besides, versions of current algorithms can be created for several fields, such as quantum neural networks, quantum deep learning, quantum-inspired ML, and quantum-enhanced ML (Dunjko & Wittek, 2020). Furthermore, new classification techniques can be developed based on the quantum variational circuit, and enhance the performance with small depth of the variational quantum circuit. Besides, classical ML techniques can be implemented on real quantum machines, or simulators.

7.4. Machine learning based quantum annealing

New future vision research, quantum annealing with classical ML can also be used to develop new QML paradigms and solve complex optimization problems related to ML problems (Mott et al., 2017). Lately, the researchers introduced a hybrid model based on standard reinforcement DL model and quantum annealing (Ayanzadeh, Halem, & Finin, 2020). Another future direction, the hybridization of adiabatic quantum computation and ML (Liu et al., 2018; Pudenz & Lidar, 2013).

7.5. Machine learning based quantum entanglement

Quantum entanglement is one of the features of QM, which differentiates quantum computing from its counterpart (conventional computing) (Horodecki, Horodecki, Horodecki, & Horodecki, 2009; Plenio & Virmani, 2014). This phenomenon is not observed, so we need measurement methods to calculate entanglement degree. Newly, a Counted publications are used entanglement measures with DL models (Levine et al., 2019; Yang & Zhang, 2020), and ML (Abdel-Aty et al., 2020; Cheng, Wang, & Zhang, 2021).

Table 7
Quantum subroutines and their applications.

Subroutine	Applications	Refs.
Amplitude amplification /Grover's algorithm	Quantum counting and searching	Brassard et al. (1998) and Grover (1996)
Sampling	Quantum deep learning and "boson sampling"	Benedetti, Garcia-Pintos et al. (2019)
Quantum phase estimation	Quantum counting and machine learning	Brassard et al. (1998), Schuld et al. (2015b) and Shor (1999)
Quantum Matrix-inversion/HHL Algorithms	QSVM, kernel least squares, and machine learning	Harrow et al. (2009)
Quantum Fourier Transform	Shor's algorithm, Cryptography, Information processing, Communication, Discrete logarithm and Phase estimation	Coppersmith (2002)
Variational quantum circuit	Optimization, classification	Chen, Yang et al. (2020), Havlíček et al. (2019), Lu and Braunstein (2014), McClean et al. (2016) and Schuld et al. (2020)
Quantum Annealing	Eigenstate solver, machine learning, optimization, nurse scheduling, and healthcare	Mott et al. (2017) and Willsch et al. (2020)

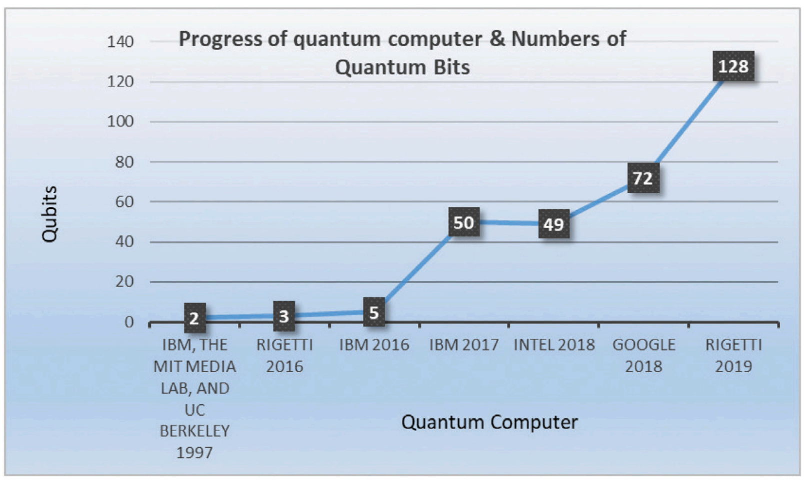


Fig. 10. Progress of quantum computers with the number of quantum bits: source, numbers of qubits achieved by different companies. The first quantum computer with two qubits was built by IBM, the MIT Media Lab, and UC Berkeley in 1997. In 2017, IBM achieved 50 qubits, and in 2019, Rigetti reached 128 qubits.

8. Conclusion

Quantum computing processes information by utilizing quantum mechanics properties such as interference, superposition, and entanglement. So, quantum computing is integrated with various fields, such as machine learning (ML), to enhance classical algorithms. This study was organized to introduce an exhaustive literature work about quantum machine learning (QML) paradigms (e.g., purely QML, hybrid classical–quantum ML, quantum-inspired ML). Besides, the most recent studies in quantum deep learning were presented. There are various methods for encoding classical data into quantum data in a Hilbert space. We discussed some of the quantum encoding methods. Many quantum machine learning using quantum subroutines were proposed to enhance the performance of classical machine learning. We discussed some of the quantum subroutines and their applications.

Future perspectives and challenges of QML were also addressed, to pave new avenues for researchers. There are still challenges for applying and implementing QML techniques with real-world problems due to the limited qubits numbers, small-scale quantum hardware, and encoding methods.

CRediT authorship contribution statement

Essam H. Houssein: Supervision, Software, Methodology, Conceptualization, Formal analysis, Investigation, Visualization, Writing – review & editing. **Zainab Abohashima:** Software, Resources, Writing – original draft. **Mohamed Elhoseny:** Methodology, Formal analysis, Data curation, Writing – review & editing. **Waleed M. Mohamed:** Methodology, Conceptualization, Formal analysis, Investigation, Data Curation, Writing – review & editing. All authors read and approved the final paper..

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

The authors would like to thank **Dr. Giuseppe Sergioli**, associate professor at the University of Cagliari (Sardinia), for informative comments and helpful discussion.

References

Abbas, A., Sutter, D., Zoufal, C., Lucchi, A., Figalli, A., & Woerner, S. (2021). The power of quantum neural networks. *Nature Computational Science*, *1*, 403–409.
Abdel-Aty, A.-H., Kadry, H., Zidan, M., Al-Sbou, Y., Zanaty, E., & Abdel-Aty, M. (2020). A quantum classification algorithm for classification incomplete patterns based on entanglement measure. *Journal of Intelligent & Fuzzy Systems*, *38*, 2809–2816.

Adhikary, S., Dangwal, S., & Bhowmik, D. (2020). Supervised learning with a quantum classifier using multi-level systems. *Quantum Information Processing*, *19*, 89.

Aïmeur, E., Brassard, G., & Gambs, S. (2006). Machine learning in a quantum world. In *Conference of the Canadian society for computational studies of intelligence* (pp. 431–442). Springer.

Albash, T., & Lidar, D. A. (2018). Adiabatic quantum computation. *Reviews of Modern Physics*, *90*, Article 015002.

Aleksandrowicz, G., Alexander, T., Barkoutsos, P., Bello, L., Ben-Haim, Y., Bucher, D., et al. (2019). Qiskit: An open-source framework for quantum computing. Accessed on: Mar, 16.

Altaisky, M. (2001). Quantum neural network. arXiv preprint quant-ph/0107012.

Alvarez-Rodriguez, U., Sanz, M., Lamata, L., & Solano, E. (2018). Quantum artificial life in an ibm quantum computer. *Scientific Reports*, *8*, 1–9.

Ambainis, A. (2012). Variable time amplitude amplification and quantum algorithms for linear algebra problems.

Amin, M. H., Andriyash, E., Rolfe, J., Kulchytskyy, B., & Melko, R. (2018). Quantum boltzmann machine. *Physical Review X*, *8*, Article 021050.

Ayanzadeh, R., Halem, M., & Finin, T. (2020). Reinforcement quantum annealing: A hybrid quantum learning automata. *Scientific Reports*, *10*, 1–11.

Barenco, A., Bennett, C. H., Cleve, R., DiVincenzo, D. P., Margolus, N., Shor, P., et al. (1995). Elementary gates for quantum computation. *Physical Review A*, *52*, 3457.

Bausch, J. (2020). Recurrent quantum neural networks. In *Advances in neural information processing systems, 33*.

Beer, K., Bondarenko, D., Farrelly, T., Osborne, T. J., Salzmann, R., Scheiermann, D., et al. (2020). Training deep quantum neural networks. *Nature Communications*, *11*, 1–6.

Benedetti, M., Garcia-Pintos, D., Perdomo, O., Leyton-Ortega, V., Nam, Y., & Perdomo-Ortiz, A. (2019). A generative modeling approach for benchmarking and training shallow quantum circuits. *Npj Quantum Information*, *5*, 1–9.

Benedetti, M., Lloyd, E., Sack, S., & Fiorentini, M. (2019). Parameterized quantum circuits as machine learning models. *Quantum Science and Technology*, *4*, Article 043001.

Bergholm, V., Izaac, J., Schuld, M., Gogolin, C., Blank, C., McKiernan, K., et al. (2018). Pennylane: Automatic differentiation of hybrid quantum–classical computations. arXiv preprint arXiv:1811.04968.

Bernstein, E., & Vazirani, U. (1997). Quantum complexity theory. *SIAM Journal on Computing*, *26*, 1411–1473.

Bettelli, S., Calarco, T., & Serafini, L. (2003). Toward an architecture for quantum programming. *The European Physical Journal D-Atomic, Molecular, Optical and Plasma Physics*, *25*, 181–200.

Biamonte, J., Wittek, P., Pancotti, N., Rebentrost, P., Wiebe, N., & Lloyd, S. (2017). Quantum machine learning. *Nature*, *549*, 195–202.

Botvinick, M., Ritter, S., Wang, J. X., Kurth-Nelson, Z., Blundell, C., & Hassabis, D. (2019). Reinforcement learning, fast and slow. *Trends in Cognitive Sciences*, *23*, 408–422.

Brassard, G., Hoyer, P., Mosca, M., & Tapp, A. (2002). Quantum amplitude amplification and estimation. *Contemporary Mathematics*, *305*, 53–74.

Brassard, G., Høyer, P., & Tapp, A. (1998). Quantum counting. In *International colloquium on automata, languages, and programming* (pp. 820–831). Springer.

Broadbent, A., & Schaffner, C. (2016). Quantum cryptography beyond quantum key distribution. *Designs, Codes and Cryptography*, *78*, 351–382.

Cao, Y., Romero, J., & Aspuru-Guzik, A. (2018). Potential of quantum computing for drug discovery. *IBM Journal of Research and Development*, *62*, 1–6.

Chakraborty, S., Shaikh, S. H., Chakrabarti, A., & Ghosh, R. (2020). A hybrid quantum feature selection algorithm using a quantum inspired graph theoretic approach. *Applied Intelligence: The International Journal of Artificial Intelligence, Neural Networks, and Complex Problem-Solving Technologies*, 1–19.

Chen, S. Y.-C., Wei, T.-C., Zhang, C., Yu, H., & Yoo, S. (2020). Quantum convolutional neural networks for high energy physics data analysis. arXiv preprint arXiv:2012.12177.

Chen, S. Y.-C., Yang, C.-H. H., Qi, J., Chen, P.-Y., Ma, X., & Goan, H.-S. (2020). Variational quantum circuits for deep reinforcement learning. *IEEE Access*, *8*, 141007–141024.

Chen, S. Y.-C., & Yoo, S. (2021). Federated quantum machine learning. *Entropy*, *23*, 460.

Cheng, S., Wang, L., & Zhang, P. (2021). Supervised learning with projected entangled pair states. *Physical Review B*, *103*, Article 125117.

Chiribella, G., D'Ariano, G. M., & Perinotti, P. (2008). Quantum circuit architecture. *Physical Review Letters*, *101*, Article 060401.

Chrisley, R. (1995). Quantum learning. In *New directions in cognitive science: Proceedings of the international symposium, Vol. 4*. Saariselka: Citeseer.

Ciliberto, C., Herbster, M., Ialongo, A. D., Pontil, M., Rocchetto, A., Severini, S., et al. (2018). Quantum machine learning: a classical perspective. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, *474*, Article 20170551.

Cleve, R., Ekert, A., Macchiavello, C., & Mosca, M. (1998). Quantum algorithms revisited. *Proceedings of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences*, *454*, 339–354.

Cong, I., Choi, S., & Lukin, M. D. (2019). Quantum convolutional neural networks. *Nature Physics*, *15*, 1273–1278.

Coppersmith, D. (2002). An approximate fourier transform useful in quantum factoring. arXiv preprint quant-ph/0201067.

Coronato, A., Naeem, M., De Pietro, G., & Paragliola, G. (2020). Reinforcement learning for intelligent healthcare applications: A survey. *Artificial Intelligence in Medicine*, *109*, Article 101964.

Cross, A. (2018). The ibm q experience and qiskit open-source quantum computing software. *APS*, *2018*, L58–003.

D-Wave (2018). D-wave releases hybrid workflow platform to build and run quantum hybrid applications in leap quantum application environment. https://www.dwavesys.com/press-releases/d-wave-releases-hybrid-workflow-platform-build-and-run-quantum-hybrid-applications/.

Dang, Y., Jiang, N., Hu, H., Ji, Z., & Zhang, W. (2018). Image classification based on quantum k-nearest-neighbor algorithm. *Quantum Information Processing*, *17*, 239.

Deutsch, D., & Jozsa, R. (1992). Rapid solution of problems by quantum computation. *Proceedings of the Royal Society of London. Series A: Mathematical and Physical Sciences*, *439*, 553–558.

DiVincenzo, D. P., Bacon, D., Kempe, J., Burkard, G., & Whaley, K. B. (2000). Universal quantum computation with the exchange interaction. *Nature*, *408*, 339–342.

Dong, D., Chen, C., & Chen, Z. (2005). Quantum reinforcement learning. In *International conference on natural computation* (pp. 686–689). Springer.

Dong, D., Chen, C., Li, H., & Tarn, T.-J. (2008). Quantum reinforcement learning. *IEEE Transactions on Systems, Man and Cybernetics, Part B (Cybernetics)*, *38*, 1207–1220.

Dunjko, V., & Briegel, H. J. (2018). Machine learning & artificial intelligence in the quantum domain: a review of recent progress. *Reports on Progress in Physics*, *81*, Article 074001.

Dunjko, V., Taylor, J. M., & Briegel, H. J. (2016). Quantum-enhanced machine learning. *Physical Review Letters*, *117*, Article 130501.

Dunjko, V., & Wittek, P. (2020). A non-review of quantum machine learning: trends and explorations. *Quantum Views*, *4*, 32.

Ezhov, A. A., & Ventura, D. (2000). Quantum neural networks. In *Future directions for intelligent systems and information sciences* (pp. 213–235). Springer.

Farhi, E., & Neven, H. (2018). Classification with quantum neural networks on near term processors. arXiv preprint arXiv:1802.06002.

Fingerhuth, M., Babej, T., & Wittek, P. (2018). Open source software in quantum computing. *PLoS One*, *13*, Article e0208561.

Freedman, M., Kitaev, A., Larsen, M., & Wang, Z. (2003). Topological quantum computation. *American Mathematical Society. Bulletin*, *40*, 31–38.

Gambs, S. (2008). Quantum classification. arXiv preprint arXiv:0809.0444.

Gao, Z., Ma, C., Song, D., & Liu, Y. (2017). Deep quantum inspired neural network with application to aircraft fuel system fault diagnosis. *Neurocomputing*, *238*, 13–23.

Green, A. S., Lumsdaine, P. L., Ross, N. J., Selinger, P., & Valiron, B. (2013). Quipper: a scalable quantum programming language. In *Proceedings of the 34th ACM SIGPLAN conference on Programming language design and implementation* (pp. 333–342).

Grover, L. K. (1996). A fast quantum mechanical algorithm for database search. In *Proceedings of the twenty-eighth annual ACM symposium on Theory of computing* (pp. 212–219).

Grzesiak, N., Blümel, R., Wright, K., Beck, K. M., Pisenti, N. C., Li, M., et al. (2020). Efficient arbitrary simultaneously entangling gates on a trapped-ion quantum computer. *Nature Communications*, *11*, 1–6.

Gupta, S., & Zia, R. (2001). Quantum neural networks. *Journal of Computer and System Sciences*, *63*, 355–383.

Hancock, A., Garcia, A., Shedenhelm, J., Cowen, J., & Carey, C. (2018). *Cirq: A Python framework for creating, editing, and invoking quantum circuits*. Google.

Hansen, B. E. (2009). *Lecture notes*, *Lecture notes on nonparametrics*.

Harrow, A. W., Hassidim, A., & Lloyd, S. (2009). Quantum algorithm for linear systems of equations. *Physical Review Letters*, *103*, Article 150502.

Havlíček, V., Córcoles, A. D., Temme, K., Harrow, A. W., Kandala, A., Chow, J. M., et al. (2019). Supervised learning with quantum-enhanced feature spaces. *Nature*, *567*, 209–212.

Henderson, M., Shakya, S., Pradhan, S., & Cook, T. (2020). Quanvolutional neural networks: powering image recognition with quantum circuits. *Quantum Machine Intelligence*, *2*, 1–9.

Herbster, M., Mountney, P., Piat, S., & Severini, S. (2020). Data encoding and classification. US Patent App. 16/265, 375.

Hidary, J. D. (2019). Development libraries for quantum computer programming. In *Quantum computing: An applied approach* (pp. 61–79). Springer.

Horodecki, R., Horodecki, P., Horodecki, M., & Horodecki, K. (2009). Quantum entanglement. *Reviews of Modern Physics*, *81*, 865.

Houssein, E. H., Abohashima, Z., Elhoseny, M., & Mohamed, W. M. (2021). Hybrid quantum convolutional neural networks model for covid-19 prediction using chest x-ray images. arXiv preprint arXiv:2102.06535.

Huang, H.-Y., Broughton, M., Mohseni, M., Babbush, R., Boixo, S., Neven, H., et al. (2021). Power of data in quantum machine learning. *Nature Communications*, *12*, 1–9.

Jeswal, S., & Chakraverty, S. (2019). Recent developments and applications in quantum neural network: a review. *Archives of Computational Methods in Engineering*, *26*, 793–807.

Jiang, T., Gradus, J. L., & Rosellini, A. J. (2020). Supervised machine learning: a brief primer. *Behavior Therapy*, *51*, 675–687.

Kamruzzaman, A., Alhwaiti, Y., Leider, A., & Tappert, C. C. (2019). Quantum deep learning neural networks. In *Future of information and communication conference* (pp. 299–311). Springer.

Khoshaman, A., Vinci, W., Denis, B., Andriyash, E., Sadeghi, H., & Amin, M. H. (2018). Quantum variational autoencoder. *Quantum Science and Technology*, *4*, Article 014001.

Killoran, N., Bromley, T. R., Arrazola, J. M., Schuld, M., Quesada, N., & Lloyd, S. (2019). Continuous-variable quantum neural networks. *Physical Review Research*, *1*, Article 033063.

Killoran, N., Izaac, J., Quesada, N., Bergholm, V., Amy, M., & Weedbrook, C. (2019). Strawberry fields: A software platform for photonic quantum computing. *Quantum*, *3*, 129.

Kubat, M. (2017). *An introduction to machine learning*. Springer.

Kuo, E.-J., Fang, Y.-L. L., & Chen, S. Y.-C. (2021). Quantum architecture search via deep reinforcement learning. arXiv preprint arXiv:2104.07715.

Lahtinen, V., & Pachos, J. K. (2017). A short introduction to topological quantum computation. *SciPost Physics*, *3*.

Lamata, L. (2020). Quantum machine learning and quantum biomimetics: A perspective. *Machine Learning: Science and Technology*, *1*, Article 033002.

LaRose, R. (2019). Overview and comparison of gate level quantum software platforms. *Quantum*, *3*, 130.

LaRose, R., & Coyle, B. (2020). Robust data encodings for quantum classifiers. arXiv preprint arXiv:2003.01695.

Laumann, C. R., Moessner, R., Scardicchio, A., & Sondhi, S. L. (2015). Quantum annealing: The fastest route to quantum computation? *The European Physical Journal Special Topics*, *224*, 75–88.

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, *521*, 436–444.

Levine, Y., Sharir, O., Cohen, N., & Shashua, A. (2019). Quantum entanglement in deep learning architectures. *Physical Review Letters*, *122*, Article 065301.

Li, R. Y., Di Felice, R., Rohs, R., & Lidar, D. A. (2018). Quantum annealing versus classical machine learning applied to a simplified computational biology problem. *NPJ Quantum Information*, *4*, 1–10.

Li, Z.-T., Meng, F.-X., Zhang, Z.-C., & Yu, X.-T. (2020). Qubits' mapping and routing for nisq on variability of quantum gates. *Quantum Information Processing*, *19*, 1–25.

Li, Y., Zhou, R.-G., Xu, R., Luo, J., & Hu, W. (2020). A quantum deep convolutional neural network for image recognition. *Quantum Science and Technology*, *5*, Article 044003.

Li, Y., Zhou, R., Xu, R., Luo, J., & Jiang, S.-X. (2020). A quantum mechanics-based framework for eeg signal feature extraction and classification. *IEEE Transactions on Emerging Topics in Computing*.

Liu, J., Spedalieri, F. M., Yao, K.-T., Potok, T. E., Schuman, C., Young, S., et al. (2018). Adiabatic quantum computation applied to deep learning networks. *Entropy*, *20*, 380.

Lloyd, S., Mohseni, M., & Rebentrost, P. (2013). Quantum algorithms for supervised and unsupervised machine learning. arXiv preprint arXiv:1307.0411.

Lu, S., & Braunstein, S. L. (2014). Quantum decision tree classifier. *Quantum Information Processing*, *13*, 757–770.

Luis, A., & Peřina, J. (1996). Optimum phase-shift estimation and the quantum description of the phase difference. *Physical Review A*, *54*, 4564.

Manzalini, A. (2019). Complex deep learning with quantum optics. *Quantum Reports*, *1*, 107–118.

Mari, A., Bromley, T. R., Izaac, J., Schuld, M., & Killoran, N. (2020). Transfer learning in hybrid classical-quantum neural networks. *Quantum*, *4*, 340.

Masanes, L., Galley, T. D., & Müller, M. P. (2019). The measurement postulates of quantum mechanics are operationally redundant. *Nature Communications*, *10*, 1–6.

McArdle, S., Endo, S., Aspuru-Guzik, A., Benjamin, S. C., & Yuan, X. (2020). Quantum computational chemistry. *Reviews of Modern Physics*, *92*, Article 015003.

McClean, J. R., Romero, J., Babbush, R., & Aspuru-Guzik, A. (2016). The theory of variational hybrid quantum–classical algorithms. *New Journal of Physics*, *18*, Article 023023.

Mehta, P., Bukov, M., Wang, C.-H., Day, A. G., Richardson, C., Fisher, C. K., et al. (2019). A high-bias, low-variance introduction to machine learning for physicists. *Physics Reports*, *810*, 1–124.

Mitarai, K., Negoro, M., Kitagawa, M., & Fujii, K. (2018). Quantum circuit learning. *Physical Review A*, *98*, Article 032309.

Moll, N., Barkoutsos, P., Bishop, L. S., Chow, J. M., Cross, A., Egger, D. J., et al. (2018). Quantum optimization using variational algorithms on near-term quantum devices. *Quantum Science and Technology*, *3*, Article 030503.

Montanaro, A. (2016). Quantum algorithms: an overview. *Npj Quantum Information*, *2*, 1–8.

Mott, A., Job, J., Vlimant, J.-R., Lidar, D., & Spiropulu, M. (2017). Solving a higgs optimization problem with quantum annealing for machine learning. *Nature*, *550*, 375–379.

Nawaz, S. J., Sharma, S. K., Wyne, S., Patwary, M. N., & Asaduzzaman, M. (2019). Quantum machine learning for 6 g communication networks: State-of-the-art and vision for the future. *IEEE Access*, *7*, 46317–46350.

Nghiem, N. A., Chen, S. Y.-C., & Wei, T.-C. (2020). A unified classification framework with quantum metric learning. arXiv preprint arXiv:2010.13186.

Pepper, A., Tischler, N., & Pryde, G. J. (2019). Experimental realization of a quantum autoencoder: The compression of qutrits via machine learning. *Physical Review Letters*, *122*, Article 060501.

Perdomo-Ortiz, A., Benedetti, M., Realpe-Gómez, J., & Biswas, R. (2018). Opportunities and challenges for quantum-assisted machine learning in near-term quantum computers. *Quantum Science and Technology*, *3*, Article 030502.

Phillipson, F., Wezeman, R. S., & Chiscop, I. (2020). Three quantum machine learning approaches or mobile user indoor-outdoor detection. In *3rd international conference on machine learning for networking (MLN'2020)(Online)*.

Plenio, M. B., & Virmani, S. S. (2014). An introduction to entanglement theory. In *Quantum information and coherence* (pp. 173–209).

Pomarico, D., Fanizzi, A., Amoroso, N., Bellotti, R., Biafora, A., Bove, S., et al. (2021). A proposal of quantum-inspired machine learning for medical purposes: An application case. *Mathematics*, *9*, 410.

Preskill, J. (2018). Quantum computing in the nisq era and beyond. *Quantum*, *2*, 79.

Pudenz, K. L., & Lidar, D. A. (2013). Quantum adiabatic machine learning. *Quantum Information Processing*, *12*, 2027–2070.

Raussendorf, R., Browne, D. E., & Briegel, H. J. (2003). Measurement-based quantum computation on cluster states. *Physical Review A*, *68*, Article 022312.

Raussendorf, R., & Harrington, J. (2007). Fault-tolerant quantum computation with high threshold in two dimensions. *Physical Review Letters*, *98*, Article 190504.

Rebentrost, P., Mohseni, M., & Lloyd, S. (2014). Quantum support vector machine for big data classification. *Physical Review Letters*, *113*, Article 130503.

Rebufello, E., Piacentini, F., Avella, A., Lussana, R., Villa, F., Tosi, A., et al. (2021). Protective measurement—a new quantum measurement paradigm: Detailed description of the first realization. *Applied Sciences*, *11*, 4260.

Rieffel, E. G., Venturelli, D., O'Gorman, B., Do, M. B., Prystay, E. M., & Smelyanskiy, V. N. (2015). A case study in programming a quantum annealer for hard operational planning problems. *Quantum Information Processing*, *14*, 1–36.

Romero, J., Olson, J. P., & Aspuru-Guzik, A. (2017). Quantum autoencoders for efficient compression of quantum data. *Quantum Science and Technology*, *2*, Article 045001.

Roncaglia, A. J., Cerisola, F., & Paz, J. P. (2014). Work measurement as a generalized quantum measurement. *Physical Review Letters*, *113*, Article 250601.

Ruan, Y., Xue, X., Liu, H., Tan, J., & Li, X. (2017). Quantum algorithm for k-nearest neighbors classification based on the metric of hamming distance. *International Journal of Theoeretical Physics*, *56*, 3496–3507.

Sagheer, A., Zidan, M., & Abdelsamea, M. M. (2019). A novel autonomous perceptron model for pattern classification applications. *Entropy*, *21*, 763.

Schuld, M. (2018). *Supervised learning with quantum computers*. Springer.

Schuld, M., Bocharov, A., Svore, K. M., & Wiebe, N. (2020). Circuit-centric quantum classifiers. *Physical Review A*, *101*, Article 032308.

Schuld, M., & Killoran, N. (2019). Quantum machine learning in feature hilbert spaces. *Physical Review Letters*, *122*, Article 040504.

Schuld, M., Sinayskiy, I., & Petruccione, F. (2014). The quest for a quantum neural network. *Quantum Information Processing*, *13*, 2567–2586.

Schuld, M., Sinayskiy, I., & Petruccione, F. (2015a). An introduction to quantum machine learning. *Contemporary Physics*, *56*, 172–185.

Schuld, M., Sinayskiy, I., & Petruccione, F. (2015b). Simulating a perceptron on a quantum computer. *Physics Letters. A*, *379*, 660–663.

Schuld, M., Sinayskiy, I., & Petruccione, F. (2016). Prediction by linear regression on a quantum computer. *Physical Review A*, *94*, Article 022342.

Sergioli, G. (2020). Quantum and quantum-like machine learning: A note on differences and similarities. *Soft Computing*, *24*, 10247–10255.

Sergioli, G., Giuntini, R., & Freytes, H. (2019). A new quantum approach to binary classification. *PLoS One*, *14*, Article e0216224.

Sergioli, G., Russo, G., Santucci, E., Stefano, A., Torrisi, S. E., Palmucci, S., et al. (2018). Quantum-inspired minimum distance classification in a biomedical context. *International Journal of Quantum Information*, *16*, Article 1840011.

Sheng, Y.-B., & Zhou, L. (2017). Distributed secure quantum machine learning. *Science Bulletin*, *62*, 1025–1029.

Shor, P. W. (1999). Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM Review*, *41*, 303–332.

Sierra-Sosa, D., Arcila-Moreno, J., Garcia-Zapirain, B., Castillo-Olea, C., & Elmaghraby, A. (2020). Dementia prediction applying variational quantum classifier. arXiv preprint arXiv:2007.08653.

da Silva, A. J., Ludermir, T. B., & de Oliveira, W. R. (2016). Quantum perceptron over a field and neural network architecture selection in a quantum computer. *Neural Networks*, *76*, 55–64.

Silverman, B. (2018). *Density estimation for statistics and data analysis*. [epub].

Simon, D. R. (1997). On the power of quantum computation. *SIAM Journal on Computing*, *26*, 1474–1483.

Sinaga, K. P., & Yang, M.-S. (2020). Unsupervised k-means clustering algorithm. *IEEE Access*, *8*, 80716–80727.

Smith, R. S., Curtis, M. J., & Zeng, W. J. (2016). A practical quantum instruction set architecture. arXiv preprint arXiv:1608.03355.

Steiger, D. S., Häner, T., & Troyer, M. (2018). Projectq: an open source software framework for quantum computing. *Quantum*, *2*, 49.

Svore, K., Geller, A., Troyer, M., Azariah, J., Granade, C., Heim, B., et al. (2018). Q# enabling scalable quantum computing and development with a high-level dsl. In *Proceedings of the real world domain specific languages workshop 2018* (pp. 1–10).

Tiwari, P., & Melucci, M. (2019). Towards a quantum-inspired binary classifier. *IEEE Access*, *7*, 42354–42372.

Von Lilienfeld, O. A. (2018). Quantum machine learning in chemical compound space. *Angewandte Chemie International Edition*, *57*, 4164–4169.

Wallnöfer, J., Melnikov, A. A., Dür, W., & Briegel, H. J. (2020). Machine learning for long-distance quantum communication. *PRX Quantum*, *1*, Article 010301.

Wang, W., Yang, N., Zhang, Y., Wang, F., Cao, T., & Eklund, P. (2016). A review of road extraction from remote sensing images. *Journal of Traffic and Transportation Engineering (English Edition)*, *3*, 271–282.

Wiebe, N., Kapoor, A., & Svore, K. M. (2014). Quantum deep learning. arXiv preprint arXiv:1412.3489.

Willsch, D., Willsch, M., De Raedt, H., & Michielsen, K. (2020). Support vector machines on the d-wave quantum annealer. *Computer Physics Communications*, *248*, Article 107006.

Yang, C.-H. H., Qi, J., Chen, S. Y.-C., Chen, P.-Y., Siniscalchi, S. M., Ma, X., et al. (2020). Decentralizing feature extraction with quantum convolutional neural network for automatic speech recognition. arXiv preprint arXiv:2010.13309.

Yang, Z., & Zhang, X. (2020). Entanglement-based quantum deep learning. *New Journal of Physics*, *22*, Article 033041.

Ying, M., Feng, Y., Duan, R., Li, Y., & Yu, N. (2012). Quantum programming: From theories to implementations. *Chinese Science Bulletin*, *57*, 1903–1909.

Zhao, Z., Pozas-Kerstjens, A., Rebentrost, P., & Wittek, P. (2019). Bayesian deep learning on a quantum computer. *Quantum Machine Intelligence*, *1*, 41–51.

Zhong, Y., & Yuan, C. (2012). Quantum competition network model based on quantum entanglement. *Journal of Computers*, *7*, 2312–2317.

Zhou, R. (2010). Quantum competitive neural network. *International Journal of Theoeretical Physics*, *49*, 110–119.

Zidan, M. (2020). A novel quantum computing model based on entanglement degree. *Modern Physics Letters B*, *34*, Article 2050401.

Zidan, M., Abdel-Aty, A.-H., El-shafei, M., Feraig, M., Al-Sbou, Y., Eleuch, H., et al. (2019). Quantum classification algorithm based on competitive learning neural network and entanglement measure. *Applied Sciences*, *9*, 1277.