Implementation of Quantum Machine Learning Algorithms: A Literature Review

Marija Šćekić
Faculty for Information Technologies
Mediterranean University
Podgorica, Montenegro
Email: marijascekic1982@gmail.com

Snežana Šćepanović
Faculty for Information Technologies
Mediterranean University
Podgorica, Montenegro
Email: snezana.scepanovic@unimediteran.net

Sandra Mitrović
Dalle Molle Institute of AI
IDSIA - USI/SUPSI
Lugano, Switzerland
Email: sandra.mitrovic@idsia.ch

2022 11th Mediterranean Conference on Embedded Computing (MECO) | 978-1-6654-6828-2/22/$31.00 ©2022 IEEE | DOI: 10.1109/MECO55406.2022.9797184

***Abstract*—Recently, machine learning is benefiting from advantages of quantum computing which has resulted in a new stream of algorithms known as quantum machine learning algorithms. This paper presents the literature describing implementations of quantum machine learning algorithms in the various quantum machine learning frameworks. In addition, for each of the observed algorithms and frameworks, the literature in which they are described is stated. To the best of our knowledge, this is so far the most comprehensive overview of the existing QML algorithms with their corresponding implementation frameworks.**

I. INTRODUCTION

Quantum computing is computing based on properties of quantum physics. It is no longer studied only theoretically but also practiced on quantum devices developed on different hardware platforms. Since it is possible to execute up to ten thousand elementary operations on the devices with fifty to a hundred qubits [1], this area is becoming more and more interesting for learning and research as well.

Thanks to the quantum entanglement and the fact that qubit does not only permit states 0 and 1 (as does the bit), quantum algorithms have a significant advantage over the classical ones in solving different kinds of problems. Quantum machine learning (QML) algorithms are super promising thanks to rapid progress of quantum technologies [2]. Given the importance of quantum algorithm implementation for the field of quantum computing [3], we created an overview of the literature that contains information on the implementation of QML algorithms. In addition, we presented the literature for each observed quantum machine learning (QML) framework and QML algorithm.

The rest of paper is organized as follows. The second chapter gives a brief introduction to quantum computing and the third chapter presents a brief introduction to QML. Fourth chapter provides an overview of literature on implementation of QML algorithms. Also, for each of the observed algorithms, the literature describing them is stated. In the fifth chapter, we provide an insight into the literature of every QML framework mentioned in the fourth chapter.

II. QUANTUM COMPUTING

Qubits play a role of bits in quantum setting and, hence, represent the essential quantum computing concept [4]. In this section, we give a brief overview of few basics terms from the field of quantum computing.

1) Qubit: A Qubit (or a quantum bit), defined in quantum system, is analog of the classical binary bit. The qubit state can be in state 0 and 1 (as in the classical system) but also in the superposition of states 0 and 1 in accordance with the basic principles of quantum mechanics. This leads to a phenomenon known as a quantum parallelism in which it is possible to count on 0 and 1 at the same time [5].

2) Quantum register: A quantum register is composed of multiple qubits. If it contains n qubits then it allows superposition of 2^n states, and thus computing 2^n states at the same time [5].

3) Entanglement: The particles are entangled if their quantum states cannot be described independently, no matter how far apart they are. Quantum teleportation, fast quantum algorithms and quantum error-correction are based on quantum entanglement [6].

4) Measurements: Measurements represent reading out of qubits and storing information about the obtained 0 or 1 into classical bits. The measurement outcomes are probabilistic according to quantum mehanic [7].

III. QUANTUM MACHINE LEARNING

Machine learning (ML) allows us to solve or get approximate solutions to many tasks. However, ML has its limitations related to long executions times and limited capacity. Quantum computing can improve ML in terms of speed of execution, efficiency in learning and an increment in the capacity of associative or content-addressable memories [8]. The possibilities provided by QML and the properties of quantum physics are used to solve ML problems. There are two main types of QMLs, namely quantum-applied and quantum-enhanced machine learning. Quantum-inspired machine learning and quantum generalized learning ideas also attract attention as lines of research [9].

IV. IMPLEMENTATION OF QML ALGORITHMS

In the Table I, we present the literature overview on the implementation of the QML algorithms, that we encountered in the literature together with the corresponding frameworks. QML algorithms are shown as rows and QML frameworks are shown as columns. Additionally, in this chapter we present a very brief description of the algorithms mentioned in Table I and the literature in which they are described.

978-1-6654-6828-2/22/$31.00 ©2022 IEEE

TABLE I

A literature review on implementation of QML algorithms (rows) and QML frameworks (columns). Implementations presented in scientific articles are marked with '*' while implementations provided in web resources of corresponding frameworks are marked with '**'.

	Qiskit	**Strawberry Fields**	**TensorFlow Quantum**	**Amazon Braket**	**PennyLane**	**Microsoft QDK**	**Qibo**	**Forest**
QKA	[10]**							
GBS		[3]*			[47]**			
CV-IQP		[3]*						
CV GSA		[3]*						
QSVM-Kernel	[14]*		[14]*	[14]*				
VarQBM	[41]*		[15]*					
VQC	[42]**, [48]**				[16]**			
QSVM	[43]**							
QGANs	[44]**				[39]**			
VQLS	[45]**				[46]**			
CCQC					[49]**			
AAVQE							[4]*	
QKS								[21]*
QK-medians	[22]*							
QK-means	[23]*							
QNN	[24]**							
CVQNN		[51]**						
QCNN			[50]**					
QvNN					[52]**			

1) Quantum kernel alignment (QKA): QKA is a classical-quantum algorithm that executes steps in iterations. Parameterized quantum circuits for evaluating the quantum kernel matrices with quantum kernel estimator (QKE) are executed by quantum hardware. The parameters of these circuits are adapted by classical optimizer to maximize the alignment [10].

2) Gaussian Boson Sampling (GBS): GBS uses experimental resource of Single Mode Squeezed States (SMSS) to implement a Boson Sampling [11].

3) Continuous-variable instantaneous quantum polynomial (CV-IQP): CV-IQP is the extension of the Instantaneous Quantum Polynomial (IQP) to the continuous-variable domain [12], [3].

4) Continuous-variable Grover's search algorithm (in our notation: CV GSA): In the classical system, O(N) steps are required to search a list of N elements that are not sorted. In Grover's quantum search algorithm $O(\sqrt{N})$ steps are required to perform such search [13]. In [13], this (originally proposed for discrete-variable quantum systems) algorithm was generalised to continuous variables.

5) The support vector machine with a quantum kernel estimator (QSVM-Kernel): A quantum machine learning algorithm QSVM-Kernel was suggested to solve classification problems. QSVM-Kernel method can replace the classical feature space in realistic physics datasets using great dimensionality of the quantum Hilbert space [14].

6) Variational Quantum Boltzmann Machines (VarQBM): VarQBM is based on variational quantum algorithms. VarQBM is compatible with generic Hamiltonians and the NISQ era quantum hardware [15].

7) Variational Quantum Classifier (VQC): Quantum circuits Variational Quantum Classifiers can be trained from marked data to classify new data samples [16]. VQC can be more efficient using far fewer free parameters than the classic model [17].

8) Quantum SVM algorithm (QSVM): Quantum SVM algorithm is used to solve classification problems that require a feature map (for which computing the kernel in classical way cannot be done efficiently) [18].

9) Quantum generative adversarial networks (QGANs): A very interesting novelty in deep neural networks is generative adversarial network (GANs) [19]. QGANs have potential exponential acceleration in terms of performance [2].

10) Variational Quantum Linear Solver (VQLS): In the last decade, attention has been paid to the possibilities of solving linear equations using quantum computers, because they play an important role in the fields of science and technology. Variational Quantum Linear Solver is a quantum-classical algorithm for solving linear systems [20].

11) Circuit-centric quantum classifiers (in our notation: CCQC): With circuit-centric classifiers the main idea is to use a generic quantum circuit composed of one and two-qubit quantum gates as a classification model [1].

12) Adiabatically assisted VQE (AAVQE): AAVQE is an extension of the Variation Quantum Eigensolver (VQE) algorithm that may be useful for improving optimization [4].

13) Quantum Kitchen Sinks (QKS): Quantum Kitchen Sinks is a QML algorithm that does not require expensive parameter optimization of quantum circuit. The idea for creating this algorithm came from a technique known as random kitchen sinks. Optimizing machine learning tasks can be much simpler with random nonlinear transformations [21].

14) Quantum k-medians algorithm using parallel Euclidean distance estimator (in our notation: QK-medians): QK-medians distance estimator has an exponential speed up compared to the classical k-medians algorithm [22].

15) Quantum k-means algorithm (in our notation: QK-means): The classical K-means algorithm has the bottleneck that is the calculation of the distance between N-dimensional vectors. QK-means provides an exponential speed up of calculating distances using the quantum parallelism [23].

16) Quantum Neural Networks (QNN): Different QNNs available in Qiskit Machine Learning are discussed in [24], such as NeuralNetwork, OpflowQNN, TwoLayerQNN and CircuitQNN.

17) Quantum Continuous-variable neural network (CVQNN): Another version of quantum neural networks, where fully connected neural network layer has been implemented using continuous-variable model (more precisely, variational quantum circuit) has been implemented in [25].

18) Quantum Convolutional neural network (QCNN): QCNN, proposed in [26], is an adaptation of classical Convolutional Neural Networks to the quantum domain, using a quantum circuit model to implement constituting layers (convolutional and pooling).

19) Quanvolutional neural network (in our notation QvNN): Compared to convolutional neural networks, quanvolutional neural networks have an added quantum convolutional (quanvolutional) layer consisting of quantum filters that are based on random quantum circuits [27].

V. Quantum machine learning frameworks

In this chapter we present a very brief description of the frameworks mentioned in Table I and the literature in which they are described. A literature review of some of those frameworks can be also find in various literature e.g. [5], [28].

1) Qiskit: Qiskit is an open-source software development kit (SDK) that allows working with quantum computers at the level of pulses, circuits, and application modules [29]. Qiskit's Machine Learning package contains datasets suitable for ML problems and several classification algorithms such as QSVM and VQC that can use that data for experiments [30].

2) Strawberry Fields: Strawberry Fields is an open-source software architecture for photonic quantum computing. It is a full-stack library (built in Python) for design, simulation, optimization, and QML of continuous variable circuits [31].

3) TensorFlow Quantum: In early 2020 Google announced a new library for quantum machine learning TensorFlow Quantum (TFQ). Only Python language is used for implementation and current support [32]. TensorFlow Quantum is an open source library created for the speedy prototyping of hybrid classical-quantum models either for quantum or classical data. TFQ enables high-level abstractions for handling discriminative and generative quantum models. Also, high-performance quantum circuit simulators are supported by TFQ [33].

4) Amazon Braket: Amazon Braket is a fully managed quantum computing service, whose design is adapted to contribute to the acceleration of scientific research and software development in the field of quantum computing [34].

5) PennyLane: PennyLane is a cross-platform Python library. It is designed for differentiable programming of quantum computers [35]. Basically, it is designed for machine learning techniques in quantum computers [32].

6) Microsoft QDK: Microsoft's Quantum development kit (QDK) with Q# (its programming language) is ready to empower education and research in quantum computing/programming. The resources estimator provided with the quantum development kit allows profiling of quantum algorithms, so researchers can estimate the exact costs of quantum algorithms and their applications for commercial purposes [36].

7) Qibo: Qibo is an open-source software that allows quick evaluation quantum circuits and adiabatic evolution which takes full advantage of hardware accelerators. [4].

8) Forest: Forest SDK is a set of software tools that we can use to write quantum programs. Using Quantum Cloud Services (QCS) or a simulator, we can can be compile and run them [37].

VI. Discussion

Several different insights can be obtained from Table I regarding the state of the quantum machine learning algorithms implementation:

- Qiskit ([29]) seems to be the most popular platform for QML algorithm implementation (even when considering only scientific articles).
- Most of the papers focus on implementing a single QML algorithm for a specific QML framework.
- There is only one paper ([14]) that implements the same algorithm for different QML frameworks.

We have also noted a couple of other interesting aspects:

- Some papers focus on a single QML algorithm but implement it in different variants. As such, different variants of the QK-means algorithm are considered in [23].
- Although rare, there exist algorithm implementations coded using two different frameworks (QML framework combined with another, more general-purpose, framework for quantum programming). Such example is the implementation of VarQBM in [15] performed using a combination of TensorFlow Quantum and Cirq, an open source framework for programming quantum computers [38]. Another example is the implementation of QGANs in [39], performed using a combination of PennyLane and Cirq.
- With respect to the availability of the code, literature can be divided into three categories: literature providing full implementation code, literature providing code snippets and finally, literature just mentioning implementation without providing any coding details.
- Even at the current stage, QML provides quantum implementations of machine learning algorithms that range from traditional ones, such as SVM and K-means/medians, to the some of the most recent deep learning architectures, such as GANs.

VII. Conclusion

In this paper, we presented the literature that contains information on the implementation of QML algorithms. For the observed algorithms and frameworks, we have also stated the literature describing them.

To the best of our knowledge, this is so far the most comprehensive overview of the existing QML algorithms with their corresponding implementation frameworks. It can be noted that there is a wide diversity in the choice of implementation platforms for QML algorithms. While it might be interesting to provide missing implementations of QML algorithms in different frameworks, for the future work we suggest a study to review and compare quantum machine learning frameworks in terms of their usability for introductory level of education in the field of QML, as is done for the quantum software development kits (QSDKs) [40].

REFERENCES

[1] M. Schuld, A. Bocharov, K. Svore, N. Wiebe, *"Circuit-centric quantum classifiers,"* https://arxiv.org/abs/1804.00633.

[2] T. Li, S. Zhang1, J. Xia, *Quantum Generative Adversarial Network: A Survey* ,Computers, Materials & Continua, vol.64, no.1, pp.401-438, 2020.

[3] D. Su, K. K. Sabapathy, C. R. Myers, H. Qi, C. Weedbrook, K. Brádler *"Implementing quantum algorithms on temporal photonic cluster states,"* https://arxiv.org/abs/1805.02645.

[4] S. Efthymiou, S. Ramos-Calderer, C. Bravo-Prieto, A. P´erez-Salinas, D. Garc´ıa-Mart´ın, A. Garcia-Saez, J. I. Latorre, S. Carrazza *Qibo: a framework for quantum simulation with hardware acceleration*, https://arxiv.org/abs/2009.01845.

[5] F. Ablayev, M. Ablayev, J. Zhexue Huang, K. Khadiev, N. Salikhova, D. Wu,*"On Quantum Methods for Machine Learning Problems Part I: Quantum Tools,"* Big data mining and analytics, ISSN 2096-0654 04/06 pp 41–55 Volume 3, Number 1, March 2020, DOI: 10.26599/BDMA.2019.9020016.

[6] M. A. Nielsen, I. L. Chuang, *"Quantum Computation and Quantum Information,"* Cambridge, UK: Cambridge University Press, 2010.

[7] J. Abhijith, A. Adedoyin, J. Ambrosiano, P. Anisimov, A. Bartschi, W. Casper, G. Chennupati, C. Coffrin, H. Djidjev, D. Gunter, S. Karra, N. Lemons, S. Lin, A. Malyzhenkov, D. Mascarenas, S. Mnisyewski, B. Nadiga, D. O'Mallly, D. Oyen, S. Pakin, L. Prasad, R. Roberts, P. Romero, N. Santhi, N. Sinitsyn, P. J. Swart, J. G. Wendelberger, B. Yoon, R. Zamora, W. Zhu, S. Eidenbenz, P. J. Coles, M. VuFffray, A. Y. Lokhov, *"Quantum Algorithm Implementations for Beginners,"* available at: https://arxiv.org/abs/1804.03719.

[8] N. Neumann, F. Phillipson, R. Versluis, *"Machine learning in the quantum era,"* Article in Digitale Welt, April 2019, DOI: 10.1007/s42354-019-0164-0.

[9] T. M. Khan, A. Robles-Kelly, *"Machine Learning: Quantum vs Classical,"* IEEE ACCESS DOI: 10.1109/ACCESS.2020.3041719.

[10] *Quantum Kernel Alignment with Qiskit Runtime* available at: https://qiskit.org/documentation/partners/qiskit_ibm_runtime/tutorials/qka.html

[11] C. S. Hamilton, R. Kruse, L. Sansoni,S. Barkhofen, C. Silberhorn, I. Jex *"Gaussian Boson Sampling,"* https://arxiv.org/abs/1612.01199.

[12] T. Douce, D. Markham, E. Kashefi, E. Diamanti, T. Coudreau, P. Milman, P. van Loock, G. Ferrini *"Continuous-Variable Instantaneous Quantum Computing is hard to sample,"* https://arxiv.org/pdf/1607.07605.pdf.

[13] A. K. Pati, S. L. Braunstein, S. Lloyd *"Quantum searching with continuous variables,"* https://arxiv.org/pdf/quant-ph/0002082.pdf.

[14] S. L. Wu, S. Sun, W. Guan, C. Zhou, J. Chan, C. L. Cheng, T. Pham, Y. Qian, A. Z. Wang, R. Zhang *"Application of Quantum Machine Learning using the Quantum Kernel Algorithm on High Energy Physics Analysis at the LHC,"* https://arxiv.org/abs/2104.05059.

[15] L. Lehtonen *"Analysis and implementation of quantum Boltzmann machines,"* Master's thesis, Faculty of Science University of Helsinki , available at https://helda.helsinki.fi/bitstream/handle/10138/332586/Lehtonen_Leevi_thesis_2021.pdf?sequence=2&isAllowed=y.

[16] *Variational classifier* https://pennylane.ai/qml/demos/tutorial_variational_classifier.html#sphx-glr-demos-tutorial-variational-classifier-py.

[17] E. M. G. Fuster, *Variational Quantum Classifier,* , available at: http://diposit.ub.edu/dspace/bitstream/2445/140318/1/GIL%20FUSTER%20Elies%20Miquel.pdf.

[18] *Quantum SVM algorithm,*, available at: https://qiskit.org/documentation/stable/0.26/stubs/qiskit.aqua.algorithms.QSVM.html?highlight=qsvm\#qiskit.aqua.algorithms.QSVM.

[19] P. Dallaire-Demers, N. Killoran, *Quantum generative adversarial networks*, https://arxiv.org/abs/1804.08641.

[20] C. Bravo-Prieto, R. LaRose, M. Cerezo, Y. Subaşı, L. Cincio, P. J. Coles, *Variational Quantum Linear Solver* , https://arxiv.org/abs/1909.05820.

[21] C. M. Wilson, S. Otterbach, N. Tezak, R. S. Smith, A. M. Polloreno, P. J. Karalekas, S. Heidel, M. S. Alam, G. E. Crooks, M. P. Silva *Quantum Kitchen Sinks: An algorithm for machine learning on near-term quantum computers* https://arxiv.org/abs/1806.08321.

[22] A. T. Getachew *Quantum K-medians Algorithm Using Parallel Euclidean Distance Estimator,* https://arxiv.org/abs/2012.11139.

[23] S. U. KHAN *"Quantum K means Algorithm,"* Master's thesis, School of Electrical Engineering and Computer Science, KTH Royal Institute of Technology, available at http://kth.diva-portal.org/smash/get/diva2:1381305/FULLTEXT01.pdf.

[24] *Quantum Neural Networks,* available at: https://qiskit.org/documentation/machine-learning/tutorials/01_neural_networks.html.

[25] N. Killoran, T. R. Bromley, J. M. Arrazola, M. Schuld, N. Quesada, S. Lloyd *Continuous-variable quantum neural networks,* available at: https://arxiv.org/abs/1806.06871.

[26] I. Cong, S. Choi, M. D. Lukin *Quantum Convolutional Neural Networks,* https://arxiv.org/pdf/1810.03787.pdf.

[27] M. Henderson1, S. Shakya1, S. Pradhan1, T. Cook1 *Quanvolutional Neural Networks: Powering Image Recognition with Quantum Circuits,* available at: https://arxiv.org/pdf/1904.04767.pdf.

[28] D. P. García, J. Cruz-Benitob, F. J. García-Peñalvo, *"Systematic Literature Review: Quantum Machine Learning and its applications,"* available at: https://arxiv.org/abs/2201.04093

[29] *Qiskit,* availableat:https://qiskit.org.

[30] *Qiskit Machine Learning,* available at: https://github.com/Qiskit/qiskit-machine-learning.

[31] N. Killoran, J. Izaac, N. Quesada, V. Bergholm, M. Amy, C. Weedbrook *"Strawberry Fields: A Software Platform for Photonic Quantum Computing,"* available at: https://arxiv.org/abs/1804.03159.

[32] *"Exploring Information for Quantum Machine Learning Models,"* https://ir.library.louisville.edu/cgi/viewcontent.cgi?article=4808&context=etd.

[33] M. Broughton, G. Verdon, T. McCourt, A. J. Martinez, J. H. Yoo, S. V. Isakov, P. Massey, R. Halavati, M. Y. Niu, A. Zlokapa, E. Peters, O. Lockwood, A. Skolik, S. Jerbi, V. Dunjko, M. Leib, M. Streif, D. V. Dollen, H. Chen, S. Cao, R. Wiersema, H. Huang, J. R. McClean, R. Babbush, S. Boixo, D. Bacon, A. K. Ho, H. Neven, M. Mohseni. *"TensorFlow Quantum: A Software Framework for Quantum Machine Learning,"* https://arxiv.org/pdf/2003.02989.pdf.

[34] *Amazon Braket* , available at: https://aws.amazon.com/braket/.

[35] *Pennylane* https://pennylane.ai.

[36] *Quantum Development Kit, Q#, and Katas,* https://www.microsoft.com/en-us/research/video/quantum-development-kit-q-and-katas/.

[37] *Forest,* , available at: https://docs.rigetti.com/qcs/.

[38] *Cirq,* available at: https://quantumai.google/cirq.

[39] *Quantum Generative Adversarial Networks with Cirq + TensorFlow,*, available at: https://pennylane.ai/qml/demos/tutorial_QGAN.html.

[40] M. Šćekić, A. Yakaryılmaz, *"Comparing Quantum Software Development Kits for Introductory Level Education,"* , Baltic J. Modern Computing, Vol. 10 (2022), No. 1, pp. 87–104, available at: https://doi.org/10.22364/bjmc.2022.10.1.06.

[41] C. Zoufal, A. Lucchi, S. Woerner. *"Variational quantum Boltzmann machines,"* https://arxiv.org/pdf/2006.06004.pdf

[42] *"Variational Quantum Classifier (VQC),"* https://qiskit.org/documentation/stable/0.24/tutorials/machine_learning/03_vqc.html.

[43] *Quantum-enhanced Support Vector Machine (QSVM),* available at: https://qiskit.org/documentation/stable/0.24/tutorials/machine_learning/01_qsvm_classification.html.

[44] *Quantum Generative Adversarial Networks*, available at: https://learn.qiskit.org/course/machine-learning/quantum-generative-adversarial-networks.

[45] *Variational Quantum Linear Solver,*, available at: https://qiskit.org/textbook/ch-paper-implementations/vqls.html.

[46] *Variational Quantum Linear Solver,*, available at: https://pennylane.ai/qml/demos/tutorial_vqls.html.

[47] *Quantum advantage with Gaussian Boson Sampling,* available at: https://pennylane.ai/qml/demos/tutorial_gbs.html.

[48] *Building a Variational Quantum Classifier,* available at: https://www.qmunity.tech/tutorials/building-a-variational-quantum-classifier.

[49] *Inside Quantum Classifiers,* available at: https://github.com/microsoft/QuantumKatas/blob/main/tutorials/QuantumClassification/InsideQuantumClassifiers.ipynb.

[50] *Quantum Convolutional Neural Network,* available at: https://www.tensorflow.org/quantum/tutorials/qcnn.

[51] *Quantum neural network,* available at: https://strawberryfields.ai/photonics/demos/run_quantum_neural_network.html#id7.

[52] *Quanvolutional Neural Networks,* available at: https://pennylane.ai/qml/demos/tutorial_quanvolution.html.