International Journal of Theoretical Physics (2020) 59:4013–4024
https://doi.org/10.1007/s10773-020-04656-1

Quantum Driven Machine Learning

Shivani Saini [1] · PK Khosla [1] · Manjit Kaur [1] · Gurmohan Singh [1]

Received: 1 June 2020 / Accepted: 10 November 2020 / Published online: 1 December 2020

Abstract
Quantum computing is proving to be very beneficial for solving complex machine learning problems. Quantum computers are inherently excellent in handling and manipulating vectors and matrix operations. The ever increasing size of data has started creating bottlenecks for classical machine learning systems. Quantum computers are emerging as potential solutions to tackle big data related problems. This paper presents a quantum machine learning model based on quantum support vector machine (QSVM) algorithm to solve a classification problem. The quantum machine learning model is practically implemented on quantum simulators and real-time superconducting quantum processors. The performance of quantum machine learning model is computed in terms of processing speed and accuracy and compared against its classical counterpart. The breast cancer dataset is used for the classification problem. The results are indicative that quantum computers offer quantum speed-up.

Keywords Qubit · Quantum computing · Machine learning · Support vector machine · Big data

1 Introduction

The size of datasets has grown rapidly owing to boom of internet-of-things (IoT) devices and systems like smartphones, sensor networks for smart-cities, wireless sensor networks, remote sensing, and radio-frequency identification (RFID) readers, and smart home etc. The global data volume has increased from 4.4 zettabytes in 2013 to 44 zettabytes in 2020 [1]. It has been further predicted that by 2025, the global data volume will reach 163 zettabytes [1]. This exponential increase in data volume has given birth to a new term called big data. Big data field basically deals with new methods of extracting meaningful information from complex or

✉ Gurmohan Singh
gurmohan@cdac.in

1 Centre for Development of Advanced Computing (C-DAC), Mohali, Ministry of Electronics & Information Technology (MeitY), Mohali 160071, India

voluminous datasets and analyzing them which are very difficult to be processed with conventional methods, softwares, and hardware computing resources [2].

The developments in quantum computing have provided a tool to handle big data related problems in various fields like machine learning, artificial intelligence, molecular modeling, chemistry, optimization etc. Basically, quantum computing is a revolutionary change in terms of way of processing the data. The classical Turing machines are based on two states *True* and *False* which are represented by Boolean bits *1* and *0,* respectively whereas quantum computers are capable of manipulating 1, 0, and all linear combinations of *1* and *0* states simultaneously. The basic entity of representing state of quantum systems is called quantum bit (Qubit). The principles of superposition and entanglement provide quantum computers immense power of handling and manipulating many quantum states simultaneously [3]. Quantum computing prominences on the difficulty of store, process, and transfer of information that is encoded in quantum mechanical systems. In other words, it is the process of manipulating the information using quantum circuits/systems or algorithms. The mode of information is known as quantum information [4].

The quantum gates are generally expressed as unitary matrices. They are categorized into single and multi qubits gates. By using single qubit quantum gates, we are able to deploy basic unitary transformations like phase and amplitude of qubit. Pauli's X, Y, and Z-gates are widely used gates for phase and amplitude transformations. The Hadamard *(H)* gate is used for creating superposition of interacting qubits [4] whereas controlled NOT (CNOT) gate is used to represent entanglement in quantum circuits and systems.

The Table 1 illustrates single-qubit gates, their corresponding unitary transformation matrices and type of mapping they perform on the inputs. Similarly, the Table 2 illustrates multi-qubit gates, their unitary transformation matrices and type of mapping they perform on the inputs.

Machine learning (ML) is process of enabling a computing machine to learn from data to predict the output instead of explicitly programming it to perform the same task. In statistics and artificial intelligence ML algorithms manipulate big datasets for image/speech recognition, pattern identification or strategy optimization etc. [5]. Due to the increased computational power and data availability, machine learning algorithms have become a powerful solution and

Table 1 Single-Qubit Gates [4, 21]

Single Qubit Gates	**Unitary Transformation Matrix**	**Transformation/Mapping**
Hadamarad (H) Gate	$\frac{1}{\sqrt{2}}\begin{pmatrix}1 & 1\\ 1 & -1\end{pmatrix}$	Transform finite quantum state of Qubits to superposition state
Pauli X-Gate	$\begin{pmatrix}0 & 1\\ 1 & 0\end{pmatrix}$	Invert only amplitude of Qubit i.e. $\|1\rangle \rightarrow X\|1\rangle = \|0\rangle$ and $\|0\rangle \rightarrow X\|0\rangle = \|1\rangle$
Pauli Y-Gate	$\begin{pmatrix}0 & -i\\ i & 0\end{pmatrix}$	Invert amplitude of Qubit as well as rotate phase vector of Qubit around y-axis i.e. multiply it by $\pm i$
Pauli Z-Gate	$\begin{pmatrix}1 & 0\\ 0 & -1\end{pmatrix}$	rotate phase vector of Qubit around z-axis by π
S-Gate	$\begin{pmatrix}1 & 0\\ 0 & i\end{pmatrix}$	rotate phase of Qubit state $\|1\rangle$ by $\pi/2$ around z-axis
T-Gate	$\begin{pmatrix}1 & 0\\ 0 & e^{i\Pi/4}\end{pmatrix}$	rotate phase of a Qubit state by $\pi/4$ around z-axis

Table 2 Multi-Qubit Gates [4, 21]

Multi Qubit Gates	Unitary Transformation Matrix	Transformation/Mapping
SWAP-Gate	$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$	Performs swapping of inputs and provides them as outputs
C-NOT	$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$	Performs flipping of second Qubit value if the first Qubit value is *1*

find the applications in understanding and controlling quantum systems. The quantum computational devices promise quantum speedup of machine learning algorithms for the many computational problems [6].

In Section 2 of this paper, firstly we have introduced the fields of classical and quantum machine learning and then previous works of researchers in field of quantum machine learning. In section 3, we have discussed about detailed set-up of experiments performed in this study. In Section 4 results of simulations on classical and quantum back-ends are enumerated along with their comparison. The paper concludes in Section 5.

2 Machine Learning

2.1 Classical Machine Learning

A machine can learn by two methods- learning from data and learning from interaction. There are broadly four classes of learning namely supervised, unsupervised, reinforcement, and deep learning [5–7]. Learning based on the data mining and data analysis is supervised and unsupervised learning, whereas learning based on interaction falls under reinforcement learning which enhances sequentially at every step [7]. In supervised learning, the machine concludes a function from training dataset of labelled points. The main aim is to conclude relationship between input and output and guess the output for new data or input values. The prediction probability distribution function consists of three steps of model selection, learning, and inference. The algorithm in unsupervised learning is assigned data without labels. The training set contains a set of input values. The goal is to find hidden structure in unlabelled information from input data values. Other types of unsupervised problems are clustering, dimensionality reduction. Three step process is involved in unsupervised learning model selection, learning and generation of new samples [7, 8]. Reinforcement learning is a middle ground between supervised and unsupervised learning because there is no immediate correct output to the input but there is some kind of supervision. In place of receiving the desired output for each input, it receives feedback from the environment. This helps an algorithm as feedback tells how the steps are were chosen helped or harmed the output [8]. Deep learning is basically based on artificial neural networks (ANNs). Deep learning architectures typically comprise of recurrent neural networks, deep neural networks, and convolutional neural networks. They are widely used in many applications such as natural language processing, drug design, computer vision, speech recognition, bioinformatics, and medical image analysis etc. [8].

Support vector machine (SVM) utilizes a hyperplane which act as demarcating line between two classes of data as depicted in Fig.1. The target is to find a hyperplane having maximum distance from the support vectors which are data points nearest to hyperplane. The optimization of hyperplane's position and alignment in SVM is possible using support vectors. The data in SVM is classified into two classes having values of '1' and '−1' [9].

Assume training data$\{ (x_1,\ v_1).\ .\ \ldots\ldots\ldots\ldots(x_n,\ v_n) \}$, where $v \in \{ 1,\ -1\}$ are two different classes labelled with values of '1' and '-1'. Generally, a hyperplane is represented as $w\ ^T x - b = 0$, where $\overrightarrow{w}$ is a vector normal to hyperplane and b is bias parameter. The maximum distance between support vectors of two classes is $\left\|\frac{2}{w}\right\|$. The linear SVM classifier [9] produces a decision output for a new data vector $\overrightarrow{x}_b$ is as

$$v_i\ \left(w\ ^T x_i - b\right) \geq 1 \text{ for } i = 1\ldots\ldots.N \tag{1}$$

2.2 Quantum Machine Learning

Quantum machine learning is the interaction between quantum computing and machine learning to solve complex problems that are very hard for classical machine learning [7–9]. The implementation of quantum machine algorithms basically involves supervised and unsupervised learning processes. In quantum clustering technique, quantum Lloyd's algorithm is used to solve k-means clustering problems [10]. To obtain the distance of the centroid of the cluster, a repetitive procedure is employed. In comparison with the classical machine learning algorithms quantum algorithms have an advantage of speed up the process. Quantum decision tree technique [11] is used to create the classifiers in machine learning which employs quantum states. The quantum decision tree learns from set of training datasets, where the training dataset is splitted into subsets by each node. Quantum decision tree classifies the data from the root to the ending vital leaf. The leaf assignment to a class is dependent on the target attribute state. The linear algebra based

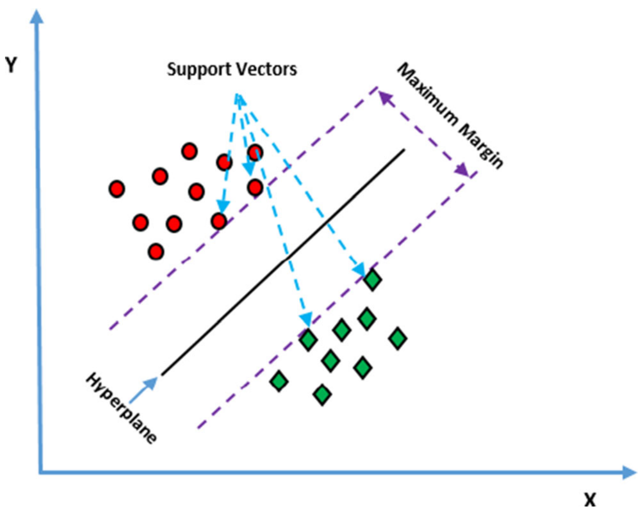


Fig. 1 Concept of hyperplane and support vectors

quantum ML comprises of data analysis and protocols of machine learning operated by unitary transformations on vectors [6, 9]. The key factor is the quantum state of n qubits in a 2^n-dimensional complex vector space. Unitary operations performed on qubits multiplies the state vector by 2^n x 2^n matrices. Quantum computers perform linear algebra operations such as Fourier transform, eigenvectors and eigenvalues [6, 9].

Quantum machine learning provides a vast scope in the techniques of computing. The principles of entanglement and superposition provide it an edge over classical machine learning. It finds applications in data handling, pattern classification, image classification, quantum clustering techniques, finding features, and quantum support vector machines etc.

Many research articles in the area of quantum machine learning focus on algorithms used for quantum machine learning. Jacob Biamonte *et. al* [7] explained machine learning for benchmarking, control and harnessing of quantum effects and also explained quantum algorithmic developments that led to speedups for computational complexity. They further discussed classical learning in quantum systems, quantum-enhanced learning for big data analytics using different algorithms like quantum support vector machine based on matrix inversion along with principal component analysis. Authors concluded that quantum computing and machine learning are enabling technologies for each other. Nimish Mishra *et. al* [8] highlighted the effects of quantum computation-based approach on normal processing and machine learning. They elaborated that quantum machine learning algorithms which were only theoretically possible in past are now available in a demonstrable form and anticipated that these algorithms could provide higher efficiency and speed in execution time. They have discussed different quantum machine learning algorithms such as SVM, Quantum SVM, and Quantum HHL algorithm etc. Authors also listed challenges of quantum machine learning algorithms like quantity of data that it could handle, problems in reading data, and issues such as reading operation cost exceeding the cost of quantum algorithm. Carlo Ciliberto *et. al* [12] delineated various machine learning techniques highlighting emerging challenge of increasing computational cost. They described linear algebra for quantum computation with its computational advantages. Vedran Dunjko *et. al* [13] enumerated links between quantum computation and quantum information along with artificial intelligence and machine learning. Various quantum algorithms like principal component analysis, quantum SVM were discussed which provides speedups for learning type problems. Maria Schuld *et. al* [6] explained the future theory of quantum learning. In quantum machine learning, there are quantum gates that are used for transformations that map quantum states onto others. Author has presented different types of quantum versions like quantum versions of k- nearest neighbor methods used for pattern classification, quantum SVM for linear discrimination, and clustering for unsupervised learning. Vojtech Havlicek *et. al* [14] implemented and proposed two novel methods on the superconducting processors. In first method, the quantum variational algorithm develops a variational quantum circuit for classification task. In the second method, a kernel based classifier guesses the kernel functions and optimize it directly.

Quantum SVM algorithm can be implemented using two different approaches on quantum machines. The first approach make use of Grover's search algorithm and provides quadratic speedup [15] while the second approach make use of HHL algorithm [16] and provide exponential speedup [17]. The HHL algorithm is capable of extracting certain attributes of $\overrightarrow{x}$ full filling requirement $A\ \overrightarrow{x} = \ \overrightarrow{b}$ where, A is an N x N matrix while $\overrightarrow{b}$ is a vector of size N × 1. Using least square approximation of SVM illustrated in [18], the quadratic problem can be mapped to a linear equation system as

$$F\begin{pmatrix} b \\ \overrightarrow{\alpha} \end{pmatrix} \equiv \begin{pmatrix} 0 & \overrightarrow{I}^{T} \\ \overrightarrow{I} & K+\gamma^{-1}I \end{pmatrix}\begin{pmatrix} b \\ \overrightarrow{\alpha} \end{pmatrix} = \begin{pmatrix} 0 \\ \overrightarrow{v} \end{pmatrix} \tag{2}$$

where, $\overrightarrow{v}$ denotes training data labels and $I$is unit matrix. The factors $\overrightarrow{\alpha}$and $b$play most important role in determining the value of SVM classifier. Here, linear kernel matrix (K) of $M \times M$ is taken and factor γ control the trade-off between SVM classifier goal and training error. Any new data vector input $\overrightarrow{x}_b$can be classified solving linear equation system of (2) as

$$v\left(\overrightarrow{x}_b\right) = \text{sgn}\left(\overrightarrow{w}\ \overrightarrow{x}_b + b\right) = \text{sgn}\left(\sum_{i=1}^{N}\alpha_i\ k\left(\overrightarrow{x}_i\ \overrightarrow{x}_b\right) + b\right) \tag{3}$$

where,$\overrightarrow{x}_i$ is training data and the $\alpha_i$represents i^{th} dimension of α.

The classical SVM classifier solves a quadratic problem in time$O(\log(\varepsilon^{-1})poly(N, M)$, where, $M$is number of training vectors, ε presents precision or correctness, and $N$is dimensionality index. The quantum SVM classifier solves a linear system of equations in time of $O(\log_2(NM))$only. Hence, quantum SVM are exponentially faster than its classical counterparts [17]. The other reason behind exponential speedup of information processing on quantum computers may be attributed to fact that quantum states stored in quantum random access memory (QRAM) are accessible parallelly [17, 19].

3 Experimental Methods

The proposed experiment uses both classical and quantum SVM learning algorithms for breast cancer classification problem. The Wisconsin Diagnostic dataset for breast cancer is utilized in this experiment [20]. This dataset has 569 observations with 32 columns that are attributes for the prediction of the diagnosis value of two types of cancer one is benign and another type is malignant. The different attributes present in data provide information about the unique ID number of the patients (ID number), diagnosis of the two types of cancer that are M = malignant or B = benign, computed mean, standard and worst errors for features such as radius, area, compactness, texture, perimeter, smoothness, concave points, concavity, symmetry, and fractional dimension. The Fig. 2 illustrates the IBM Qiskit based simulation framework used in this work.

The experimental results are computed in terms of accuracy and execution time to demonstrate quantum speedup provided by quantum computational approach. The computational models are implemented on different back-ends including simulator installed on local machine and real-time IBM simulator and superconducting quantum processors [21]. The libraries used in classical machine learning as well as quantum machine learning includes NumPy, Pandas, and Matplotlib, etc. The main difference in the quantum machine learning working is the inclusion of a special quantum library called as QISKIT provided by IBM [22]. In this study, data classification algorithms such as the classical and quantum support vector machine algorithm are used. SVM is a type of supervised learning algorithm widely used regression and classification [9]. But, in the proposed framework it is used for classification between the type of cancer named *benign* and *malignant*. The SVM algorithm basically is based upon the idea of finding a hyperplane or a line that could possibly divide the whole datasets into two or any given number. Support vectors are the data points or values which are nearest to the

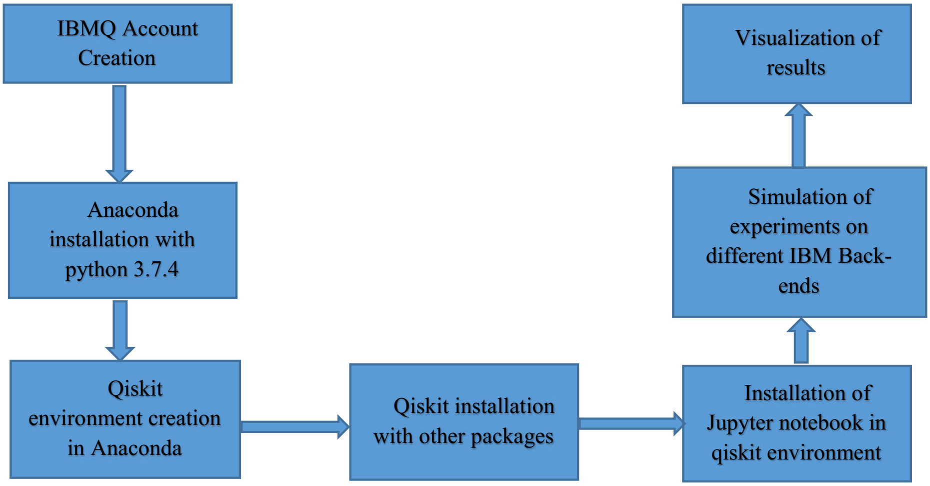


Fig. 2 Qiskit based simulation framework

hyperplane, removing or altering them would affect the classification line (hyperplane). The problem arises when there is no clear hyperplane which would divide the data, then it is necessary to move away from 2-dimensional (2D) to 3-dimensional view (3D) [17]. When the data is very complex it requires more dimensional planes or feature spaces, this becomes really hard for classical computers to compute. To compute complex data quantum support vector kernel algorithm combines quantum computing and machine learning to efficiently classify things on higher dimensional [17].

For executing classical machine learning algorithm, we need a Local CPU environment while for qiskit is used for running quantum support vector machine algorithm. The qiskit has three parts namely the provider, the backend, and the job. The provider provides access to different types of backends. There are two providers named *Aer* and IBMQ [22]. *Aer* provides access to simulators within qiskit that can run on your local machine e.g. *qasm_simulator statevector_simulator, unitary_simulator*, and *clifford_simulator*. IBMQ extends access to the cloud-based back-ends. The simulators and real quantum devices will be hosted on IBMQ [21]. The back-ends signifies either a simulator or a real quantum processor used for the running of the quantum circuit and generating results. The job finds out the state of execution i.e. whether the job is queued, running, or failed. Four IBMQ backends have been used in this experiment named *ibmq_qasm_simulator, ibmq_16_melbourne, ibmqx2*, and *qasm_simulator* is used for a variational approach for support vector machine algorithm. *ibmq_qasm_simulator* is a 32-qubit simulator, *ibmq_16_melbourne* and *ibmqx2* are real-time superconducting 14 and 5-qubit processors, respectively.

The Table 3 lists the configurations of three types of back-ends and their features. The number of qubits each back-end support is represented by feature n_qubits and status message tell whether a back-end is active or not. Backend_version tells about version status. The basic gates used in a particular back-end are described by feature basic_gate. The other features are maximum shots (max_shots) and maximum_experiments allowed in a particular back-end. Figure 3 describes implementation flow of classical and quantum machine learning models.

Table 3 Features of IBM back-ends [21, 22]

Properties	ibmq_qasm_ simulator	ibmq_16_ melbourne	ibmqx2
n_qubits	32	14	5
status_msg	active	active	active
backend_version	0.1547	2.0.6	2.0.5
basic_gates	*'u1', 'u2', 'u3', 'cx', 'cz', 'id', 'x', 'y', 'z', 'h', 's', 'sdg', 't', 'tdg', 'ccx', 'swap', 'unitary', 'initialize', 'kraus'*	*'id', 'u1', 'u2', 'u3', 'cx',*	*'u1', 'u2', 'u3', 'cx', 'id'*
max_shots	8192	8192	8192
max_experiments	300	75	75

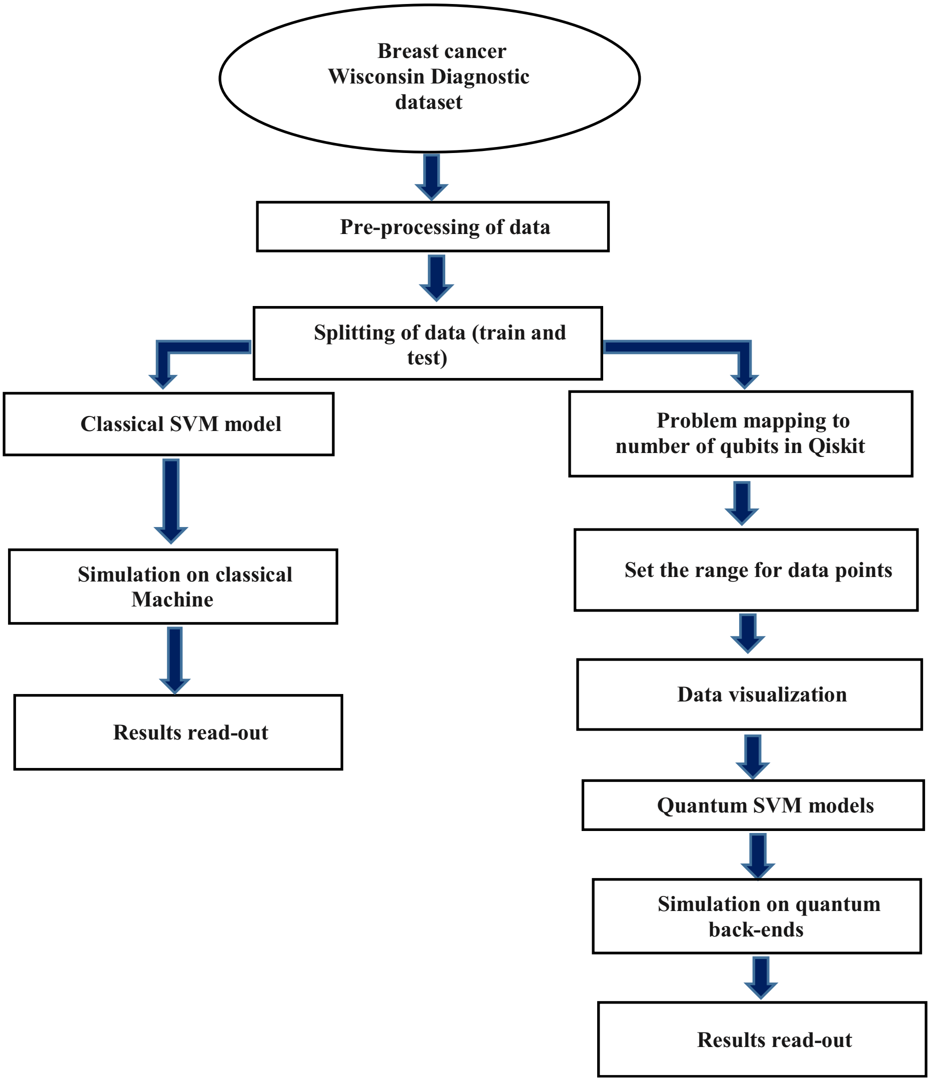
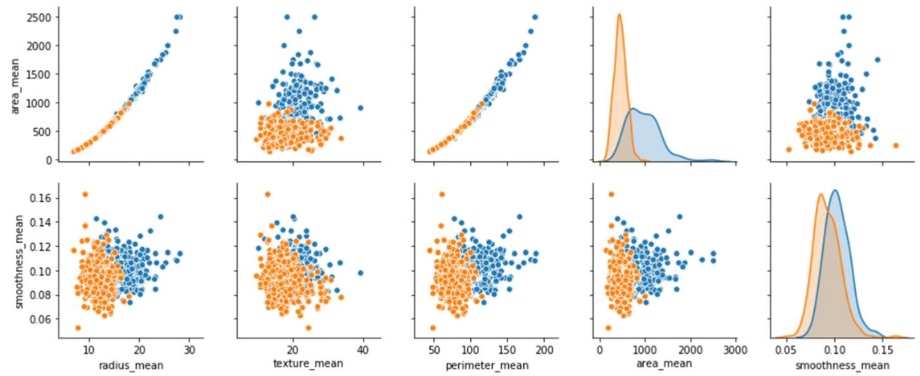


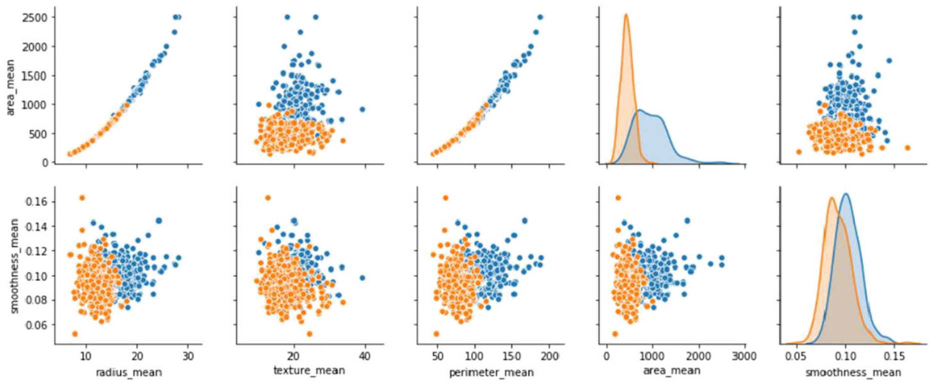
Fig. 3 Implementation of classical and quantum machine learning models

4 Results and Discussions

The proposed experiment was performed employing qiskit in jupyter notebook and python 3.7.4 for writing code for the models in this study. In this supervised learning experiment, the SVM algorithm and labelled data have been used for breast cancer classification problem. Firstly, we compute relationship between features of dataset during data pre-processing. The Fig. 4 (a) and (b) illustrates plots of relationship among five parameters of breast tumour after applying PCA and classical SVM on training data set. The radius_mean is mean of distances to the perimeter points from the center, texture_mean is standard deviation of a gray-scale value, perimeter_mean is mean size of a central tumor, area_mean is mean of the area of tumor, and smoothness_mean is the radius of the value of a mean of local variation [20, 23]. From the digitized image of a fine needle aspirate (FNA) of a breast mass, these features are computed. Authors in [24] described the characteristics of a cell nuclei in the image. FNA is a biopsy which is used by the doctors, that uses a very thin needle to extract a fluid from a suspicious area. Then, the sample of biopsy is verified to get whether there are cancer cells present in it or not [24].

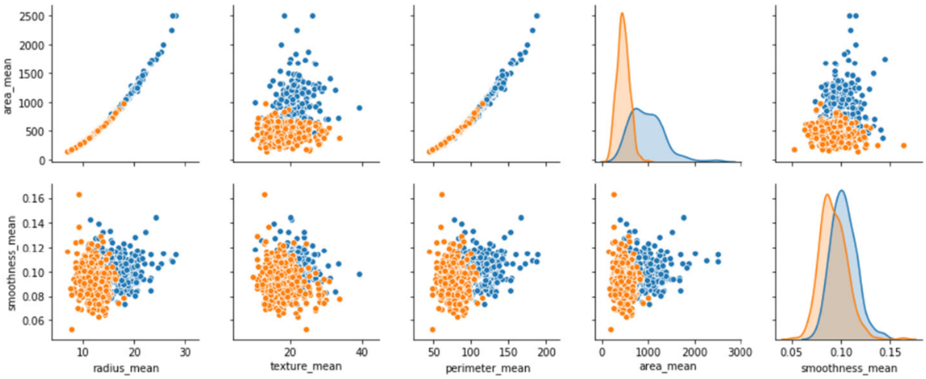


(a)

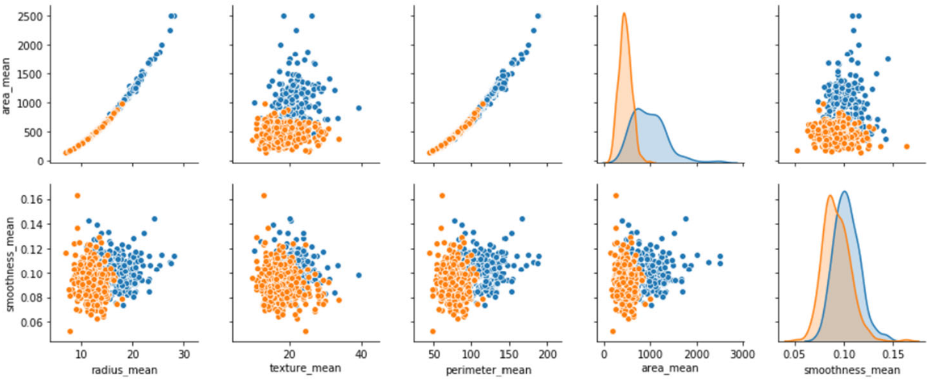


(b)

Fig. 4 Visualization of the relationship between features (a) after applying classical PCA, and (b) after applying classical SVM



(a)



(b)

Fig. 5 Visualization of the relationship between features (a) after applying quantum PCA, and (b) after applying quantum SVM

The Fig. 5 (a) and (b) illustrates plots of relationship among five parameters of breast tumour after applying quantum PCA and quantum SVM on breast cancer training data set.

The Table 4 enumerates the accuracy and execution time results obtained after running classical and quantum support vector machine-based machine learning models for classification problem using breast cancer Wisconsin datasets in this study. The quantum variational SVM based algorithm-based model is computationally more intensive as two different quantum algorithms are to be run in it. As two algorithms are to be run in it, the accuracy as well as execution time are degraded. As compared to classical SVM, the execution time is degraded 92% as indicated by downward arrow in Table 4 and accuracy is decreased by 11%. And for the quantum kernel based SVM the accuracy is 85% with three back-ends degraded by 5%. The time taken by *ibmq_qasm_simulator* is just 410 ms as it is run on local machine lesser than classical SVM by 234 folds. In case of ibmq_16_melbourne 14 qubits processor, the time taken to run the kernel based SVM model for breast cancer classification is lesser by 51.6% as compared to classical SVM model run on local machine. For 5-qubit ibmqx2 processor, the time taken to solve the classification problem is reduced by 7.5% as compared to classical SVM model.

Table 4 Comparison of performance parameters of classical and quantum models

Algorithm Used	Backend	Accuracy (%)	Execution Time (s)	% improvement/degradation in time compared with classical model
Quantum kernel based SVM	ibmq_qasm_simulator	85	0.401	99.57
	ibmq_16_melbourne	85	45.5	51.6
	ibmqx2	85	87.00	7.5
Quantum Variational SVM	qasm_simulator	80	1176	92
Classical SVM	Local CPU environment	90	94.07	

5 Conclusion

This paper practically demonstrates implementation of quantum machine learning model based on quantum support vector machine algorithm to solve classification problem using breast cancer Wisconsin datasets. The observations in this paper reveals that quantum driven machine learning could provide quantum speedup to solve many hard problems. The quantum machine learning problem solved using quantum simulator on local machine proved to be 234 folds faster than its classical counterpart. In case of 14 qubits real-time superconducting processor, the time taken to run the kernel based SVM model for breast cancer classification is reduced by 51.6% as compared to classical SVM model run on local machine. However, the accuracy is degraded by 5% for quantum model as compared to classical model which may be due to well established and standardized procedures followed in implementing classical SVM model. The results anticipate that quantum computers could very soon surpass solving feature mapping or classification problems for larger data sets than even classical supercomputing machines.

Acknowledgments Authors are highly grateful to IBMQ for providing access to their software and hardware computational resources through cloud network.

Compliance with Ethical Standards

Declaration Authors declare that there are no conflict of interests.

References

1. D. Reinsel, J. Gantz, and J. Rydning, "The Digitization of the World from Edge to Core," *IDC White Paper #US44413318*, 2018
2. Tsai, C., Lai, C., Chao, H., et al.: Big data analytics:a survey. Journal of Big Data. **2**(21), 1–32 (2015)
3. P. A. M. Dirac, "The Principles of Quantum Mechanics," 4th edition, Oxford University Press, (1930/1958)
4. Nielsen, M.A., Chuang, I.L.: Quantum Computation and Quantum Information. *Cambridge: Cambridge University Press* (2000)
5. Murphy, K.P.: Machine Learning: a Probabilistic Perspective. MIT Press (2012)
6. Schuld, M., Petruccione, I.S.F.: An introduction to quantum machine learning. Contemp. Phys. **56**(2), 172–185 (2015)
7. Biamonte, J., Wittek, P., Pancotti, N., Rebentrost, P., Wiebe, N., Lloyd, S.: Quantum Machine Learning. Nature. **549**(7671), (November 2016)
8. Mishra, N., Kapil, M., Rakesh, H., Anand, A., Warke, A., Sarkar, S., Dutta, S., Gupta, S., Dash, A.P., Gharat, R., Chatterjee, Y., Roy, S., Raj, S., Jain, V.K., Bagaria, S., Chaudhary, S., Singh, V., Maji, R.,

Dalei, P., Behera, B.K., Mukhopadhyay, S., Panigrahi, P.K.: Quantum Machine Learning : A Review and Current Status Quantum Machine Learning, pp. 1–24. https://doi.org/10.13140/RG.2.2.22824.72964

9. Wittek, P.: Quantum machine learning. Academic Press. **73-83**, (2014). https://doi.org/10.1016/B978-0-12-800953-6.00020-7
10. Lloyd, S.P.: Least squares quantization in pcm. IEEE Transactions on InformationTheory. **28**, 129–137 (1982)
11. Lu, S., Braunstein, S.: Quantum decision tree classifier. Quantum Inf. Process. **13**, 757–770 (2014). https://doi.org/10.1007/s11128-013-0687-5
12. Ciliberto, C., Herbster, M., Ialongo, A.D., Pontil, M., Rocchetto, A., Severini, S., Wossnig, L.: Quantum machine learning: A classical perspective. *Proc. R. Soc. A Math. Phys. Eng. Sci*. **474**(2209), (2018)
13. Dunjko, V., Briegel, H.J.: Machine learning & artificial intelligence in the quantum domain: A review of recent progress. *Reports Prog. Phys*. **81**(7), (2018)
14. Havlicek, V., Corcoles, A.D., Temme, K., Harrow, A.W., Kandala, A., Chow1, J.M., Gambetta, J.M.: Supervised learning with quantum-enhanced feature spaces. Nature. **567**(7747), 209–212 (2019)
15. Anguita, D., Ridella, S., Rivieccio, F., Zunino, R.: Quantum optimization for training support vector machines. Neural Netw. **16**(5–6), 763–770 (2003)
16. Harrow, A.W., Hassidim, A., Lloyd, S.: Quantum algorithm for linear systems of equations. *Physical review letters*. **103**(15), 150502 (2009)
17. Patrick Rebentrost, Masoud Mohseni, and Seth Lloyd "Quantum support vector machine for big data classification," *Phys. Rev. Lett.*, 113:130503, September 2014
18. Suykens, J.A., Vandewalle, J.: Least squares support vector machine classifiers. Neural. Process. Lett. **9**(3), 293–300 (1999)
19. Seth Lloyd, Masoud Mohseni, and Patrick Rebentrost "Quantum algorithms for supervised and unsupervised," machine learning. *arXiv:1307.0411*, July 2013
20. Breast Cancer Wisconsin (Diagnostic) Dataset. [Online]. Available: *https://www.kaggle.com/uciml/breast-cancer-wisconsin-data*
21. *IBM, IBM Quantum Experience.* https://quantumexperience.ng.bluemix.net/qx/experience. Accessed March, 12, 2020
22. *IBM, Quantum information science kit.* https://qiskit.org/aqua. accessed March, 2020
23. Adebola Lamidi, "Breast Cancer Classification Using Support Vector Machine (SVM)", Nov 23, 2018. Available:*https://towardsdatascience.com/breast-cancer-classification-using-support-vector-machine-svm-a510907d4878*
24. Fine Needle Aspiration (FNA) Biopsy of the Breast. The American Cancer Society medical and editorial content team. Available:*https://www.cancer.org/cancer/breast-cancer/screening-tests-and-early-detection/breast-biopsy/fine-needle-aspiration-biopsy-of-the-breast.html*