Open Access | Rapid and quality publishing

International Journal of
Electrical and Electronics Research (IJEER)
Research Article | Volume 10, Issue 2 | Pages 375-380 | e-ISSN: 2347-470X

Performance Analysis of Quantum Classifier on Benchmarking Datasets

Tarun Kumar[1], Dilip Kumar[2] and Gurmohan Singh[3]

[1]PhD scholar, Department of ECE, SLIET, Longowal, Punjab, India, tarunkumar2992@gmail.com
[2]Professor, Department of ECE, SLIET, Longowal, Punjab, India, dilip.k78@gmail.com
[3]Joint director, CSTD, Centre for Development of Advanced Computing (C-DAC), Mohali, India, gurmohan@cdac.in

***Correspondence:** Tarun Kumar, tarunkumar2992@gmail.com

ABSTRACT- Quantum machine learning (QML) is an evolving field which is capable of surpassing the classical machine learning in solving classification and clustering problems. The enormous growth in data size started creating barrier for classical machine learning techniques. QML stand out as a best solution to handle big and complex data. In this paper quantum support vector machine (QSVM) based models for the classification of three benchmarking datasets namely, Iris species, Pumpkin seed and Raisin has been constructed. These QSVM based classification models are implemented on real-time superconducting quantum computers/simulators. The performance of these classification models is evaluated in the context of execution time and accuracy and compared with the classical support vector machine (SVM) based models. The kernel based QSVM models for the classification of datasets when run on IBMQ_QASM_simulator appeared to be 232, 207 and 186 times faster than the SVM based classification model. The results indicate that quantum computers/algorithms deliver quantum speed-up.

Keywords: QML; Iris species; pumpkin seeds; QSVM; SVM; quantum computing.

ARTICLE INFORMATION
Author(s): Tarun Kumar, Dilip Kumar and Gurmohan Singh
Special Issue Editor: Dr. S. Gopalakrishnan;
Received: 10/05/2022; **Accepted**: 06/06/2022; **Published**: 30/06/2022;
e-ISSN: 2347-470X;
Paper Id:0422SI-IJEER-2022-21;
Citation: 10.37391/IJEER.100252
Webpage-link:
https://ijeer.forexjournal.co.in/archive/volume-10/ijeer-100252.html

This article belongs to the Special Issue on **Intervention of Electrical, Electronics & Communication Engineering in Sustainable Development**

Publisher's Note: FOREX Publication stays neutral with regard to jurisdictional claims in Published maps and institutional affiliations.

1. INTRODUCTION

Modern computers can process and manipulate vast quantity of data because of developments made in their architecture which makes them computationally fast. Big datasets must be efficiently handled and managed by classical computers in order to support today's complicated applications [1]. But in terms of performance and processing, the expanding size of data is posing significant hurdles for classical computers [2]. The rise of big data has forced the development of a new computer architecture or method for dealing with complex big data challenges. Developments in quantum computing and quantum-inspired machine learning techniques promise quantum speedups over their classical equivalents [3-4]. In fact, even before quantum computers were available, researchers began developing quantum machine learning algorithms that can outperform classical machine learning algorithms in terms of speed. All of these examples demonstrate that quantum machine learning algorithms have the potential to give significant speedups over their traditional counterparts [6-7]. The quantum computing paradigm is expected to ease the processing of large datasets and provide solutions to many complex problems. Quantum computers/algorithms are also expected to be capable of searching unsorted large databases [8], factoring numbers [9], and speedily extracting the needed patterns. They are capable of simultaneously searching for various data items and only detecting patterns of relevance [10]. Machine learning, artificial intelligence, big data analytics, financial modelling, molecular modelling, and other applications would all benefit greatly from the quantum computing revolution even before truly quantum solutions become available [11]. Machine learning (ML) and data analytics are benefitting from quantum-inspired algorithms [12]. Machine learning is expected to gain the most from advancements in quantum computing. [13] The key to success lies in translating real-world issues into quantum space.

Artificial intelligent systems can generate results with precision, provided the training of bigger datasets with machine learning algorithms. The accuracy with which data is classified based on its specific traits or features determines how well AI systems work [14]. Quantum computers have the potential to extract computationally complex data attributes, which might disclose previously unknown concepts. The researchers have shown that quantum supremacy is approaching sooner than expected [15]. Typically, machine learning comes into play when there is no methodology for tackling complex problems and large datasets with various variables. Machine learning has emerged as a major technique for dealing with big datasets in domains such as computer vision, natural language processing, computational banking, image processing, and computational biology etc. The ML algorithms extract patterns from the data to provide a more accurate perception, which aids in better prediction and decision-making. The application of quantum processing in machine learning is not really restricted to academics but industrial sector is also excited about it.

FOREX Publication
Open Access | Rapid and quality publishing
International Journal of
Electrical and Electronics Research (IJEER)
Research Article | Volume 10, Issue 2 | Pages 375-380 | e-ISSN: 2347-470X

QML applications will likely be employed to deliver more effective solutions to typical machine learning tasks in near future.

Quantum computing represents a paradigm shift in the way data is processed. Quantum computers represent information in states $|1\rangle$, $|0\rangle$, and any linear combinations of $|1\rangle$ and $|0\rangle$ states concurrently, whereas traditional computers represent the information in Boolean bits 1 and 0 [16]. The fundamental unit of expressing the state of a quantum system is referred to Quantum-bit (Qubit). Quantum computers can handle and manipulate numerous quantum states concurrently owing to the concepts of superposition and entanglement [13]. In quantum computing, quantum algorithms/circuits are required for information manipulation. Unitary matrices are commonly used to describe quantum gates. Pauli X, Y, and Z-gates perform amplitude and phase transformations. Superposition is attained by applying Hadamard gate on qubits whereas CNOT gate perform entanglement operations on qubits [17-18].

The paper is divided into the following sections: Section-2 introduces the fundamentals of classical machine learning. The quantum machine learning and quantum support vector machine algorithms are discussed in Section 3. Section 4 contains the results of the experiment as well as a comparison of conventional and QML techniques. The paper is concluded in Section 5.

2. CLASSICAL MACHINE LEARNING

A machine learns using two approaches, data-driven learning and interaction-driven learning. Machine learning may be divided as supervised [19], unsupervised [20], and reinforced [21] ML. Machine learning based data mining and data analysis is classified as both supervised and unsupervised, whereas interaction-based learning is classified as reinforcement learning, which improves progressively at each stage [5]. To grasp the notion of machine learning, a dataset $X=\{x_1, x_2, x_3, \dots, x_n\}$ is used, where x_i signifies the number of datapoints in the dataset. Dataset X is split into two parts, labelled training data (X_T) and unlabeled test data (X_0). The supervised machine learning uses a set of already established training data X_T composed of already categorized datapoints to produce a set of classifications $Y=\{y_1, y_2, y_3, \dots, y_n\}$,where y_i is the class for datapoints x_i. Both X_T and Y are put into a machine learning system that optimizes their internal parameters until the training data is categorized into the closest Y value. When the machine has fully learned, it is given input X_0 to classify, and the system predicts the output for X_0 [19]. Problems related to regression and classification are usually handled by supervised ML algorithms. In the case of unsupervised ML, the classification class is not specified, implying that Y does not exist. In such circumstances, machine learns on the basis of underlying structure of input data [22]. Unsupervised learning algorithms use training data X_T (unlabeled for unsupervised ML) as input and search for hidden structures. These algorithms aid in the problems related to clustering and dimensionality reduction tasks [5]. The unsupervised learning model involves a three-step process - Select, learn, and generate new samples [2]. Reinforced ML is at the intersection of supervised and unsupervised ML as the instant correct output to input is unavailable, but there exists some sort of supervision. It gets feedback from the environment instead of obtaining the expected outcome for each input. This aids an algorithm since feedback indicates if the steps chosen facilitated or damaged the outcome [22].

For classification problems, ML provides a variety of algorithms such as Support Vector Machines (SVM) [23], Naive bayes [24], K-nearest neighbors [25], and Decision tree [26], etc. SVM is a supervised ML algorithm which finds a hyperplane between two classes with the maximum margin between their support vectors. The maximum margin aids SVM's classification efficiency [23]. Support vectors improve the location and orientation of hyperplanes in SVM. In SVM, the data is divided into two classes with values *'-1'* and *'1'*. A training data $\{(x_1, y_1), (x_2, y_2), \dots\dots, (x_n, y_n)\}$, is considered such that $Y = \{1, -1\}$ are two distinct classes labelled with $-1,1$. A hyperplane depicted in *Figure 1* is usually expressed as $w^T x - b = 0$, where w is the hyperplane's vector normal and b is the bias parameter.

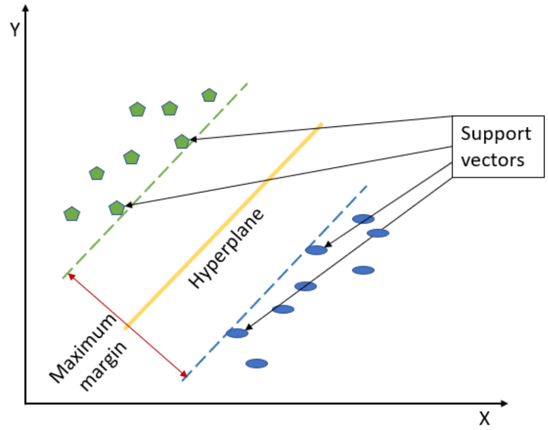


Figure 1: Representation of hyperplane and support vectors

The support vectors corresponding to both classes are at maximum distance of $\left\|\frac{2}{w}\right\|$. The decision output produced by a linear SVM classifier for new data vector x_0 is expressed as (1)

$$y_i(w^T x - b) \geq 1 \quad for\ i = 1,2,3, \dots N \qquad (1)$$

3. QUANTUM MACHINE LEARNING

Quantum machine learning combines ML and quantum computing to handle complex problems that are difficult to answer with classical ML [5,27]. In order to implement QML algorithms, supervised and unsupervised ML techniques are used. QML provides algorithms that can tackle complex issues that are difficult to address with classical ML [28]. Quantum algorithms are evolved from classical algorithms and can be implemented on quantum computers. The methods used in classical ML, such as deep neural networks, can detect statistical patterns in data and produce data that has those patterns. It has been noticed that if quantum algorithms generate statistical patterns those classical computers find difficult to generate, then quantum algorithms can identify these patterns easily [22].

Many researchers have looked at the effectiveness of Quantum Support Vector Machines for QML problems using both practical and theoretical implementations. Vedran Dunjko et al. 2018 examined various quantum algorithms such as quantum PCA and QSVM which have been numerically demonstrated to be giving quantum speed-up in ML and AI applications. Carlo Ciliberto et al. 2018 examined that the increased computational complexity and data have revolutionized ML algorithms resulting in impressive outcomes. Authors discussed about the computational cost related to the use of neural networks, linear algebra, optimization and sampling. Shivani Saini et al. 2020 implemented a QSVM based classification model on Breast cancer dataset. Authors discovered that because of computational complexity the QSVM based model results in deteriorated accuracy against the SVM but the speedup of 234 folds is delivered by quantum simulator against its classical equivalent. Gurmohan Singh et al. 2022 implemented QSVM algorithm on benchmarking MNIST dataset of pictures of handwritten numbers. Authors compared QSVM and SVM algorithms in the context of computational time and accuracy. Authors discovered that the kernel based QSVM is 81.62% computationally fast and 6.4% much accurate against the classical SVM.

The implementation of QSVM algorithm on the quantum devices can be done by two ways. The one way is to utilize the Grover's algorithm which delivers quadratic speedup [31] and other way is to utilize HHL algorithm [32] which delivers exponential speedup. The HHL algorithm efficiently extracts attributes of $\vec{x}$ satisfying the $requirement\ in\ a\vec{x} = \vec{b}\ where\ \vec{b}\ is\ a\ N\ x1\ vector\ and\ A\ is\ N\ x\ N\ matrix. SVM$ uses the least square approximation [33] which maps the quadratic problem into linear equation system and expressed as (2)

$$F\begin{pmatrix} b \\ \vec{\alpha} \end{pmatrix} = \begin{pmatrix} 0 & \vec{I}^T \\ \vec{I} & K + \gamma^{-1} I \end{pmatrix} \begin{pmatrix} b \\ \vec{\alpha} \end{pmatrix} = \begin{pmatrix} 0 \\ \vec{y_i} \end{pmatrix} \qquad (2)$$

Where I signifies unit matrix and y_i denotes the training data labels. The elements $\vec{\alpha}$ and b are most significant in defining the SVM classifier's value. A linear kernel matrix (K) of size M x M and element γ handles SVM classifier's goal and training error. For a new input data x_0, a linear equation system is used for classification and expressed as (3)

$$y_i(\vec{x_0}) = sgn(\vec{w}\,\vec{x_0} + b) = sgn\left[\sum_{i=1}^{N} \alpha_i\, k\,(\vec{x_i}\,\vec{x_0}) + b\right] \qquad (3)$$

In order to solve a quadratic problem, a classical SVM classifier takes $o(\log(\varepsilon^{-1})\, poly(N, M))$ time, where M denotes the number of training vectors, N represents dimensionality index and ε is the accuracy. The QSVM algorithm finds a solution for linear system of equations only in $o(log_2(N, M))$ time. Hence, QSVM outperforms SVM and delivers exponential speedup [10]. In addition, the quantum computers process the information and store the information on quantum RAM in the form of quantum states which can be accessed parallelly [10,28].

4. EXPERIMENTAL RESULTS

The experiment is conducted using QSVM and SVM on various benchmarking datasets to know the current position of QML in the computation field. Three datasets Iris species [34], raisin [35] and pumpkin seeds [36] are considered for this experiment. The iris species dataset comprises of three species setosa, versicolor and verginica each with 50 samples. In this dataset, one species of flower can be separated from the other two in a linear fashion, while the other two cannot. The dataset is in csv format and composed of 150 observations and 5 columns. The dataset comprises of four features i.e., sepal length, sepal width, petal length, and petal width extracted from the images of all three iris species. The aim is to classify the iris species dataset in three species. In case of raisin dataset, information of 900 raisin grains which are divided equally is present and belongs to two categories of raisins. Seven features are extracted from the images of raisins by using image processing. The features are area, minor axis length, convex area, perimeter, major axis length, eccentricity, and extent. These features provide the necessary information about all the images of raisin grains. And one column is of class in which raisin grains are to be classified. The pumpkin seeds dataset comprises of information of 2500 pumpkin seeds and divided into two classes of pumpkin seeds. The two classes of pumpkin seeds are *Urgup_Sivrisi* and *Cercevelik*. A total of 12 features are extracted from the images of pumpkin seeds. Out of 13 columns of the dataset, 12 columns belong to the features and one column belongs to class of pumpkin seeds. The extracted features are area, major axis length, convex area, eccentricity, extent, aspect ratio, perimeter, minor axis length, equip diameter, solidity, roundness and compactness. The execution time and accuracy are the performance metrics taken for the experiment which indicate the quantum simulators/computers offer quantum advantage in terms of speedup.

The QSVM based models for aforementioned datasets will be implemented on a quantum simulator [37] and a superconducting quantum computer [38,39] whereas SVM based models will be implemented on a local computer. QML algorithms make use of a quantum library QISKIT [40] for building and implementing quantum circuits and algorithms. IBM has created a QISKIT integrated platform IBM Quantum [41], which facilitates users to access their quantum simulators and real-time quantum computers through cloud. The experiment is performed on QISKIT integrated Jupiter notebook and python 3.8.8 is used for programming the models. The number of shots used for this experiment are 8192. All the datasets are split into 75% training data and 25% testing data. The kernel and variational both approaches of QSVM algorithm on datasets will be implemented on three backends ibmq_qasm_simulator [39], ibmq_quito [38] and qasm simulator [37]. Variational QSVM solve classification problems when data has more than two classes. The variational QSVM makes use of two algorithms, one for finding the hyperplane and other for classifying test data whereas kernel QSVM is based on a single algorithm and used for the tasks related to binary classification [42]. For the classification of new data, a kernel matrix is computed using quantum system and then support vectors are computed making use of classical system. The experimental setup revealed in *Table 1* comprises

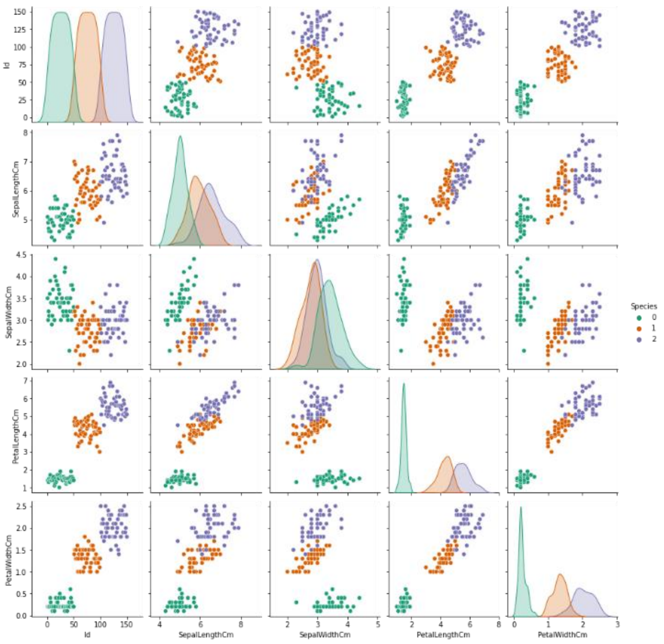
FOREX Publication
Open Access | Rapid and quality publishing
International Journal of Electrical and Electronics Research (IJEER)
Research Article | Volume 10, Issue 2 | Pages 375-380 | e-ISSN: 2347-470X

of algorithms implemented on the datasets for classification, details of backend used for the experiments and performance metrics.

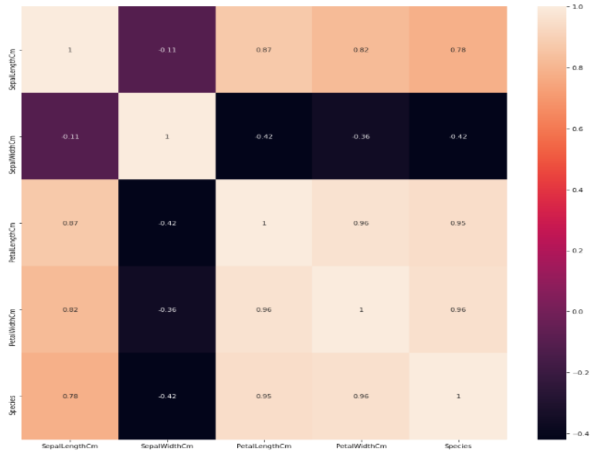
Table 1. Experimental setup of classification models

Classification algorithm	Dataset	Backend	Performance metrics
Variational QSVM	Iris species, raisin and pumpkin seeds	IBMQ Lima	Accuracy, Execution time
		IBMQ QASM simulator	
		QASM simulator	
Kernel based QSVM	Iris species, raisin and pumpkin seeds	IBMQ Lima	Accuracy, Execution time
		IBMQ QASM simulator	
		QASM simulator	
SVM	Iris species, raisin and pumpkin seeds	IBMQ Lima	Accuracy, Execution time
		IBMQ QASM simulator	
		QASM simulator	

To implement QML algorithms on all datasets, a series of steps need to be followed which includes selecting a dataset, its pre-processing, visualizing data, exploratory data analysis (EDA), data splitting, algorithm selection, dimensionality reduction using PCA, data classification using both variational and kernel based QSVM, producing quantum circuit and readout results [43]. All three datasets went through pre-processing steps like rescaling, data normalization and data cleaning etc. The algorithm selection for the datasets should be made in such a way that QML models produce valid and accurate results that EDA ensures. The datasets are now divided into training and test data followed by dimensionality reduction using PCA [44]. Lastly, SVM and QSVM (kernel/variational) are applied on the datasets for the construction of classification model and results are computed in the form of accuracy and execution time. *Figures 2-3* revealed the relationship between the features of all three datasets using heatmaps and pair plots. They are used to visualize the correlation between features of any dataset.

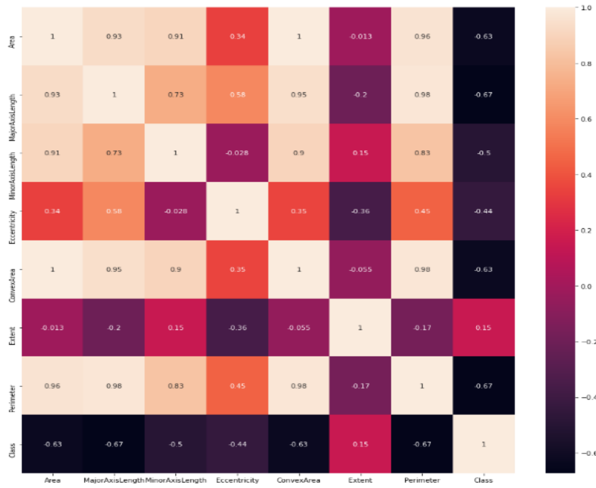


(a)

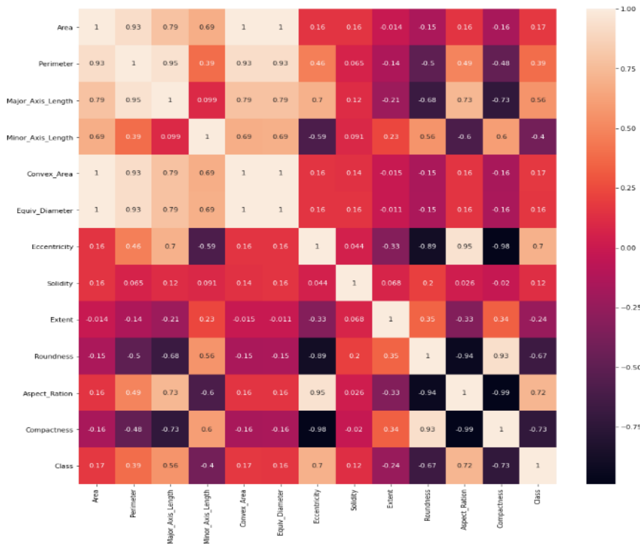


(b)

Figure 2: Visualizing **(a)** hidden information inside the features and **(b)** correlation between features of the iris species dataset



(a)



(b)

Figure 3: Visualization of correlation between the features of (a) raisin dataset and (b) pumpkin seeds dataset

Table 2 depicts the accuracy and execution time results of SVM and QSVM based classification models for three benchmarking datasets implemented on quantum computer/simulator.

Table 2. Assessment of Accuracy and execution time of classification models

Dataset	Classification Algorithm	Backend	Accuracy (%)	Execution time (s)
Iris species	Variational QSVM	IBMQ Lima	95.2	35.1
		IBMQ QASM simulator	98	0.174
		QASM simulator	98	17.6
	Kernel based QSVM	IBMQ Lima	95.2	28.6
		IBMQ QASM simulator	**98.5**	**0.112**
		QASM simulator	98	16.5
	SVM	Local CPU environment	97	26.4
Pumpkin seeds	Variational QSVM	IBMQ Lima	70.1	96.2
		IBMQ QASM simulator	73.6	0.301
		QASM simulator	71	73.4
	Kernel based QSVM	IBMQ Lima	75	84.9
		IBMQ QASM simulator	**75.3**	**0.231**
		QASM simulator	75	46.1
	SVM	Local CPU environment	83.2	48
Raisin	Variational QSVM	IBMQ Lima	61.7	124.3
		IBMQ QASM simulator	65	0.457
		QASM simulator	65	103.4
	Kernel based QSVM	IBMQ Lima	65.4	98.1
		IBMQ QASM simulator	**68**	**0.381**
		QASM simulator	68	67.2
	SVM	Local CPU environment	80	71.3

In the classification of iris species dataset, the recorded accuracy is 98% when both kernel and variational QSVM models are implemented on IBMQ_QASM_simulator and QASM_simulator. The percentage improvement of 1.5% is seen in the accuracy of variational and kernel QSVM based models against the SVM based classification model. The kernel based QSVM model takes a lot less time when compared with SVM model. The kernel based QSVM model is 232 times faster than the SVM based model for the classification of iris dataset. In case of pumpkin seeds classification, the maximum accuracy achieved by the QSVM based model is 75% when run on IBMQ_QASM_simulator. The accuracy of the SVM based model is 10% better than the kernel QSVM based classification model for pumpkin seeds dataset. But the execution time offered by the kernel based QSVM model is 207 times lesser than the SVM based classification model.

The kernel based QSVM model when implemented on the raisin dataset and run on IBMQ_QASM simulator, produce classification outcomes with 68% accuracy. The accuracy of the SVM model is 15% better than the QSVM model. But the quantum advantage in terms of speedup is seen as kernel based QSVM model for the classification of raisin dataset is 186 times faster than the SVM based classification model. The speedup advantage offered by QSVM over SVM is depicted in *Table 3*.

Table 3. Speedup offered by QSVM against SVM

Dataset	Computational speedup
Iris species	232 times
Pumpkin seeds	207 times
Raisin	186 times

5. CONCLUSION

The classification models using QSVM and classical SVM for three datasets namely iris species, raisin and pumpkin seeds has been constructed and then implemented on quantum/classical computational backends. It is found that both kernel and variational QSVM models offer speed advantage when run on the quantum backends. The kernel based QSVM models for the classification of the iris species, raisin and pumpkin seeds datasets when run on IBMQ_QASM_simulator are 232, 207 and 186 times faster than the SVM based classification model respectively. The QML algorithms take leverage of quantum mechanics principles to process multiple states simultaneously and offers speedup advantage. The kernel based QSVM model deliver results which are 1.5% more accurate than SVM model, in case of iris dataset. The percentage degradation of 10% and 15% is observed in the accuracy of the kernel based QSVM models for the classification of raisin and pumpkin seeds dataset respectively. The accuracy of theses QSVM model is less because of limited quantum data pre-processing techniques, limited hyperparameter tuning techniques, limited feature mapping techniques and noisy quantum systems. The results indicate that the QML has capability to surpass the classical ML when these issues were resolved. The speedup advantage is there but more efficient classifiers and other techniques need to be developed to improve the accuracy of the classification models.

6. ACKNOWLEDGEMENT

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

REFERENCES

[1] C. Tsai, C. Lai, H. Chao and A. V. Vasilakos, “Big data analytics: a survey,” in Journal of Big Data, vol. 2, issue no. 21, pp. 1–32, 2015.

[2] D. Fisher, R. DeLine, M. Czerwinski and S. Drucker, “Interactions with big data analytics,” Interactions, 2012. https://doi.org/10.1145/2168931.2168943.

[3] S. Saini, P. K. Khosla, M. Kaur, and G. Singh, “Quantum Driven Machine Learning,” International Journal of Theoretical Physics, vol. 59, issue no.12, pp. 4013-4024, 2020.

[4] G. Singh, M. Kaur, M. Singh and Y. Kumar, "Implementation of Quantum Support Vector Machine Algorithm Using a Benchmarking Dataset," Indian Journal of Pure & Applied Physics vol. 60, pp. 407-414, 2022.

[5] J. Biamonte, P. Wittek, N. Pancotti, P. Rebentrost, N. Wiebe and S. Lloyd, "Quantum Machine Learning." Nature, vol. 549, issue no. 7671, 2016.

[6] Y. Du, M. H..Hsieh, T. Liu, and D. Tao, "Quantum-inspired algorithm for general minimum conical hull problems," Physical Review Research, vol. 2, issue no. 3, pp. 033199, 2020.

[7] E. Tang, "Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing, (2019) https://doiorg/ 101145/33132763316310.

[8] L.K Grover, "A fast quantum mechanical algorithm for database search," Proceedings of the 28th Annual ACM Symposium on the Theory of Computation, New York: ACM Press, pp. 212–219, 1996.

[9] P.W. Shor, "Algorithms for quantum computation: discrete logarithms and factoring," Proceedings 35th Annual Symposium on Foundations of Computer Science, Santa Fe, NM, USA, 1994. https://doi.org/10.1109/SFCS.1994.365700.

[10] P. Rebentrost, M. Mohseni and S. Lloyd, "Quantum support vector machine for big data classification," Physical review letters, vol. 113, issue no.13, pp. 130503, 2014.

[11] V. Dunjko and H.J. Briegel, "Machine learning & artificial intelligence in the quantum domain," A review of recent progress. Reports Prog. Phys. Vol. 81, issue no.7, 2018.

[12] C. Ding, T.Y. Bao and H.L. Huang, "Quantum-inspired support vector machine," IEEE Transactions on Neural Networks and Learning Systems, 2021.

[13] M.A. Nielsen and I.L. Chuang, "Quantum Computation and Quantum Information," Cambridge University Press, Cambridge, 2010.

[14] M. Ying, "Quantum computation, quantum theory and AI," Artificial Intelligence, vol. 174 issue no. 2, pp. 162-176, 2010.

[15] F. Arute, K. Arya, R. Babbush, D. Bacon, J.C. Bardin, R. Barends, R. Biswas, S. Boixo, F.G. Brandao, D.A. Buell and B. Burkett, "Quantum supremacy using a programmable superconducting processor," Nature, vol. 574, issue no. 7779, pp. 505-510, 2019.

[16] E. Rieffel and W. Polak, "An Introduction to Quantum Computing for Non-Physicists," ACM Computing Surveys, vol.32, issue no. 3, pp. 300–335, 2000.

[17] M. Nagy and S.G. Akl, "Quantum computation and quantum information," Int. J. Parallel, Emergent Distrib. Syst., Vol. 21, issue no. 1, pp. 1–59, 2006.

[18] J.C. Garcia-Escartin,and P. Chamorro-Posada, "Equivalent quantum circuits," arXiv preprint arXiv:1110.2998, 2011.

[19] J. Adcock, E. Allen, M. Day, S. Frick, J. Hinchliff, M. Johnson, S. Morley-Short, S. Pallister, A. Price and S. Stanisic, "Advances in quantum machine learning," arXiv preprint arXiv:1512.02900, 2015.

[20] Z. Ghahramani, "Advanced lectures on machine learning," Journal of the Royal Statistical Society, 2004.

[21] L.P. Kaelbling, M.L. Littman and A.W. Moore, "Reinforcement learning: A survey," Journal of artificial intelligence research, vol. 4, pp. 237-285, 1996.

[22] N. Mishra, M. Kapil, H. Rakesh, A. Anand, N. Mishra, A. Warke, S. Sarkar, S. Dutta, S. Gupta, A. Prasad Dash and R. Gharat, "Quantum Machine Learning: A Review and Current Status Quantum Machine Learning," Data Management, Analytics and Innovation, pp. 1–24, 2021. https://doi.org/10.13140/RG.2.2.22824.72964

[23] A. Ukil, "Support vector machine," Intelligent Systems and Signal Processing in Power Engineering, Springer, pp. 161-226, 2007.

[24] D. Berrar, "Bayes' theorem and naive Bayes classifier. Encyclopedia of Bioinformatics and Computational Biology: ABC of Bioinformatics. Elsevier Science Publisher: Amsterdam, The Netherlands, pp. 403-412, 2018.

[25] J.M. Keller, M.R. Gray and J. A. Givens, "A fuzzy k-nearest neighbor algorithm," IEEE transactions on systems, man, and cybernetics, vol. 4, pp. 580-585, 1985.

[26] S.R. Safavian and D. Landgrebe, "A survey of decision tree classifier methodology," IEEE transactions on systems, man, and cybernetics, vol. 21 issue no. 3, pp. 660-674, 1991.

[27] P. Wittek, "Quantum machine learning," Academic Press, pp. 73-83, 2014. https://doi.org/10.1016/B978-0-12- 800953-6.00020-7.

[28] S.P. Lloyd, "Least squares quantization in PCM," IEEE Transactions on Information Theory, vol. 28, pp. 129–137, 1982.

[29] C. Ciliberto, M. Herbster, A.D. Ialongo, M. Pontil, A. Rocchetto, S. Severini, and L. Wossnig, "Quantum machine learning: A classical perspective," Proc. R. Soc. A Math. Phys. Eng. Sci., vol. 474 issue no. 2209, 2018.

[30] M. Schuld and N. Killoran, "Quantum machine learning in feature Hilbert spaces," Physical review letters, vol. 122, issue no. 4, pp. 040504, 2019.

[31] D. Anguita, S. Ridella, F. Rivieccio and R. Zunino, "Quantum optimization for training support vector machines," Neural Netw., vol. 16, issue no. 5–6, pp. 763–770, 2003.

[32] A.W. Harrow, A. Hassidim and S. Lloyd, "Quantum algorithm for linear systems of equations," Physical review letters, vol. 103, issue no.15, pp. 150502, 2009.

[33] J.A. Suykens and J. Vandewalle, "Least squares support vector machine classifiers," Neural. Process. Lett., vol. 9, issue no. 3, pp. 293–300, 1999.

[34] [dataset] https://www.kaggle.com/datasets/uciml/iris.

[35] [dataset] https://www.muratkoklu.com/datasets/

[36] IBM Inc., IBM Quantum Services. https://quantum-computing.ibm.com/services?services=simulators.

[37] IBM Inc., IBM Quantum Services. https://quantum-computing.ibm.com/services?services=systems&system=ibmq_quito.

[38] IBM Inc., IBM Quantum Services. https://quantum-computing.ibm.com/services?services=simulators&system =ibmq_qasm_simulator.

[39] IBM Inc., IBM, Quantum information science kit. https://qiskit.org/aqua.

[40] IBM Inc., IBM Quantum. https://quantum-computing.ibm.com.

[41] K. Shiba, K. Sakamoto and T. Sogabe, "Variational Quantum Support Vector Machine based on Deutsch-Jozsa Ranking," Bulletin of Networking, Computing, Systems, and Software, vol. 9, issue no. 1, pp. 63-68, 2020.

[42] G. Bologna and Y. Hayashi, "QSVM: A support vector machine for rule extraction," In International Work-Conference on Artificial Neural Networks, Springer, pp. 276-289, 2015.

[43] L.C. Wang, "Experience of data analytics in EDA and test—principles, promises, and challenges," IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 36, issue no. 6, pp. 885-898, 2016.