Experimental Comparison of Quantum and Classical Support Vector Machines

Balika J.Chelliah, ShristiShreyasi,Ananya Pandey, Kirti Singh

Abstract- Classical Support Vector Machine is hugely popular for classifying data efficiently whether it is linear or non-linear in nature. SVM has been used immensely to assist a precise classification of a data point. The kernel trick of SVM has also elevated the performance of the classical algorithm. But, SVM suffers a lot of problems on a classical machine when higher dimensions are introduced or large datasets are taken up. So, in order to enhance the efficiency of Support Vector Machine, the idea of running it on a quantum machine takes over.A Quantum Machine uses Qubits which is a single bit representing 0, 1 and superposition states of 0 & 1. This use of Qubit introduces the concept of 'parallel processing'. The Quantum Machine utilises a different version of the SVM algorithm for performing the task of classification. In the algorithm, classical data is transformed into quantum data and then analysed over a Quantum Machine. For this experiment, the outcomes from both Classical Machine as well as Quantum Machine will be compared to determine Quantum Machine's precedence over Classical Machine is justified or not. The comparison parameters are execution time and accuracy percentile of both the approaches. These results will be compared for asserting the importance of Quantum Approach in increasing machine learning's scope, application and potential for current, future and deemed impossible task.

Keywords : Quantum Machine, Quantum Machine Learning, Quantum Support Vector Machine, Support Vector Machine.

I. INTRODUCTION

Quantum Machine Learning is an interdisciplinary area which is an intersection of Quantum Physics and Machine Learning. It deals with running machine learning models on a quantum machine. A quantum machine works on qubit same as how the classical machine works on 0 and 1 bit which are expressed as high low voltage levels. A quantum bit or qubit is actually a single bit representing |0> or |1> or intermediate states of |0> and |1>. There are two concepts of quantum physics – Entanglement and Superposition which are the main basis for quantum machine learning on a quantum machine. A qubit's state comes to existence when is measured or observed otherwise it remains undefined. Quantum Machine Learning takes on two approaches- first approach involves converting classical data to quantum data then performing the computations on the quantum computer and finally converting the quantum result back into the classical format.

Revised Manuscript Received on April 07, 2019.

Dr.Balika.J.Chelliah (M.Tech, Ph.d.)-Associate Professor at CSE Department SRMIST, Ramapuram, Chennai.

Shristi Shreyasi - UG Scholar at CSE Department, SRMIST, Ramapuram, Chennai.

Ananya Pandey– UG Scholar at CSE Department, SRMIST, Ramapuram, Chennai.

Kirti Singh– UG Scholar at CSE Department, SRMIST, Ramapuram, Chennai.

In short, this methoddevelops classical ML algorithms on a quantum computer. The othermethod involves using quantum mechanical principles fordesigning machine learning algorithms for classical computers. For the contrast between a classical and a quantum machine's performance, the first approach will be used.Quantum Support Vectors Machine Algorithm uses the first approach of QML and gives precisely classified data as the end result. QSVM completes the task in an exponential time as compared to the classical SVM. It also handles the non-separable data by using a QSVM kernel algorithm which computes the kernel function much faster than the pre-existing. A decrease can be well seen in amount of time taken by the quantum algorithm as well as there is an efficient handling of large datasets for classification as it uses quantum mechanical operations. Robentrost [1], discussed about an optimised two-faced classifier which can be developed on a quantum computerhaving complexity in logarithms in the vectors size and the number of training examples. So, a quantum computer reduces a computation task taking polynomial time in the primitive environment to exponential time in a quantum environment.

II. RELATED WORK

Support Vector Machineis a supervisedalgorithm which is implemented forclassification as well as regressionproblems. But it is widelyimplemented for classification problems. In SVM, each and every data point is plotted on n-dimensional axis as a data coordinate where n stands for features held by the data with each feature acting in support for a coordinate.After plotting classification is done by finding the hyper-plane that will divide the two classes precisely.

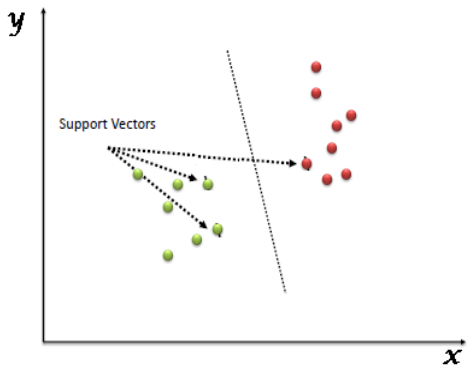


Figure 1: Classification Hyperplane of SVM

Support Vectors are coordinates of each and everydata point. SVM is aalgorithm which differentiates the two classes by a hyper-plane or line.

SVM was introduced by Vapnik [2] which stated that the classification criteria of a SVM is to find the highest margin hyper plane or differentiating line in feature space through Structural Risk Minimization that segregates with Y_j= 1 from Y_j= -1 in case of linear SVM. Thismodel finds two hyper planes present parallelly having normal vector ~u, segregated by the highest possible distance 2/|~u| separating the two classes of data. SVM is basically for linearly separable data but for non-linearly separable data, it projects the input feature space to a higher dimension of feature space which ultimately linearly separates in that higher dimension. This is inherently done by using Kernel Trick in classical SVM algorithm.

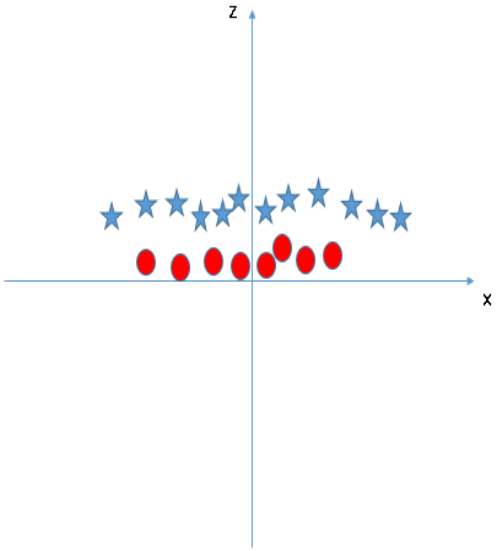


Figure 2: Projection of datapoints in n-dimensional space.

But it had drawbacks of excessive training time when large datasets were encountered as well as it was very sensitive to noise. Also, the algorithm took quadratic time for separating non-linearly separable data which demanded large memory for execution. The algorithm also suffered from heavy requirement of memory for computations. It also suffered when the input dataset is large.

Osuna [3] provided decomposition methods and Platt [4] introduced Sequential Minimal Optimisation Algorithm (SMO). They solved the problem of large datasets but they did not solve the high demand of memory. Xiao [5] provided an incremental learning SVM for combating memory requirements but it reduced the training speed of the algorithm.

Incremental SVM was introduced by Zhen [6] introduced a faster incremental SVM which curbed the problems of memory and training speed but not noise.

In this paper, we will be using Quantum Machine Learning SVM algorithm and QML SVM kernel algorithm for contrasting the difference of performance, limitation and future potential of SVM for both a classical machine run and a quantum machine in various fields.

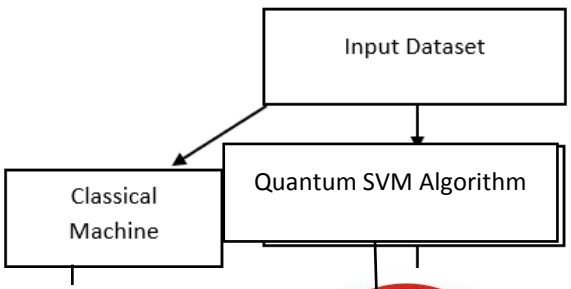
III. OVERVIEW OF THE PROPOSED SYSTEM

The proposed system curbs the vulnerabilities of the pre-existing system. A classical machine imposes many computational limitations as well on the SVM algorithm. This constraint as well the vulnerabilities discussed in the above section is completely curbed by the use of Quantum Support Vector Machine Algorithm. Robentrost [1] showed that implementation of SVM on a quantum machine gave an exponential speed up in terms of computational speed i.e. overall time complexity of the algorithm on any given dataset.Robentrost [7] also gave an optimised two-faced classifier which can be developed on a quantum computer having complexity in logarithms in the vectors size and the number of training examples.

In this, we will be using the approach of converting the classical data into quantum data before processing happen in a quantum computer via quantum support vector algorithm. The conversion into quantum data asserts the selection of number of qubits required by the quantum circuit of the quantum support vector algorithm. The qubits are single bits having values as either <0| or <1| or any state of all the superposition states possible. The state of a qubit can be known only when the bit is measured/observed.

For performing the contrast between the classical and quantum performance in terms of time complexity and accuracy, two models will be developed for each category of ML applications like pattern recognition, linear data classification, non-linear data classification, image classification etc. Both the models will also be tested on a large dataset as well as performance on a small dataset. The contrast will also be made in terms of training time and prediction time. Since a quantum computer uses the concept of parallelism i.e. a quantum computer simultaneously computer all the tasks at one time rather than computation of individual task at a single time of a classical computer. So, parallelism will also contribute to a new region of contrast between the two ways.

In the quantum approach, a quantum circuit is mainly used for implementing the algorithm on a quantum computer. This circuit uses gates like Hadamard gate H, NOT gate etc. These gates operate on the qubit selected earlier and the computations being done are represented by a unitary matrix. Llyod, Mohseni, Robentrost [8] showed that quantum computers offer an effectivestage for handling vector as well as matrices and they also give speedup in exponents over their classical counterparts in various situations like assigning N-dimensional vectors to any one of the k clusters, each one with M representative samples for which a quantum computer computes in O (log (MN)) time.



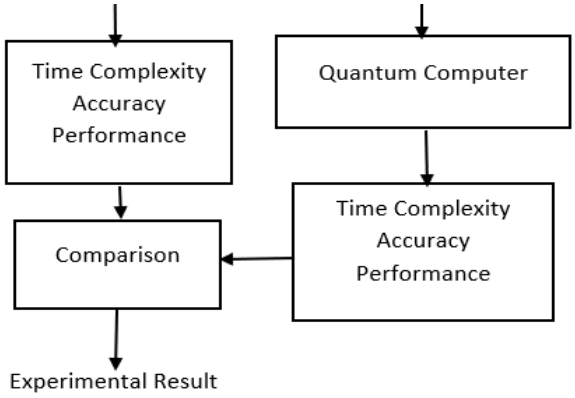


Figure 3: Flow Chart of The Proposed Model

The above flow chart shows the proposed model of the experiment. The input dataset will be pre-processed differently for both the models then fed to the respective algorithms and finally based upon time and accuracy percentile the result of the experiment will be concluded.

IV. EXPERIMENT & RESULTS

The experiment was performed on IBMQ [9] using Qiskit [10]. Scikit-Learn was used for performing the classical counterpart of the experiment. The toy datasets of Scikit-Learn such Breast Cancer (Binary class classification dataset), Iris (Multi class classification dataset) and Digits (Image Classification Dataset). These datasets had to pre-processed before feeding into quantum machine. This included setting up the range, normalizing the data, standardizing with variance and reducing no. of features to no. of qubits. The dataset pre-processing also splits the dataset into training, testing and validation/prediction dataset. In classical SVM implementation cross validation is used for validation simultaneously. Then the processed data was set up in quantum support vector machine algorithm and made to run on the real time quantum machine of IBMQ. Then accuracy percentile is taken for both the classical as well as quantum models. The execution time for various levels of the model are also taken into account. These data are compared as shown in Table 1.

Model	Dataset Pre-processing Time (s)	Train-ing Time (s)	Predic-tion Time (s)	Total Execu-tion Time (s)	Accu-racy Percen-tile
Classical SVM	1.24	0.032	0.05	1.292	0.9333
Quantum ibmqx2	0.0495	0.443	0.028	0.521	0.8965
Quantum ibmqx4	0.0448	0.407	0.005	0.457	0.8965
Quantum ibmqx_16 melbourne	0.0447	0.402	0.003	0.447	0.8965
Quantum qasm simulator	0.0174	0.400	0.002	0.419	0.8965

Table-1 Comparison of C vs. Q model

The table shows the comparison of dataset pre-processing time, training time, prediction time and accuracy percentile for classical svm as well as various qubit combinations of the quantum model. The data shown are mean data for various different datasets which were stated above. For image classification and small datasets, there was little difference between the classical and the quantum approach in terms of accuracy. But as it is observable from the Table-1, a decrease is seen in the accuracy percentile in quantum as compared to classical. This decrease can also be a result of pre-processing required to do for quantum model as well noise in the dataset and also due to the sensitivity of a quantum machine to environmental factors like temperature changes etc. There is a decrease in the execution time of models from classical svm model to ibmqx2 to ibmqx4 to ibmxq_16_melbourne to qasm simulator. QASM simulator is a backend that emulates execution of a quantum circuits on a real device and returns measurement counts. It includes highly configurable noise models and can even be loaded with automatically generated approximate noise models based on the calibration parameters of actual hardware devices.The simulator actually shows the ideal behaviour of the quantum machine on the input. Another fact to be noticed is the similarity of accuracy percentile of all the quantum models irrespective of the number of the qubits used. So, one can see that quantum model does perform better in terms of execution time but it lacks behind in accuracy.

V. CONCLUSION

In conclusion, the quantum approach for solving machine learning problems and optimising machine learning problems or algorithms has proved to be very time efficient. The execution time of classical model lags behind the execution time of the quantum model. So, one can conclude that the quantum modeldoes performs better than the classical model in terms of time complexity. This stands along with the Robentrost's [1] execution speedup in quantum model. But the quantum machine model is a bit relaxed in accuracy and preciseness of the results due to recency of the quantum field and exotic design of the quantum computer as it always requires its components to be working at near absolute zero temperature. At room temperature, the computer may acquire errors due to thermal motion of the atoms in the computer's structure. Butthese physical limitations if curbed then it is extremely advantageous to work in the direction of advancing the quantum machines for implementing various existing applications like self-driving cars, search engines, micro-processing etc and also implementing new projects on the basis of quantum computer.

REFERENCES

1. P. Rebentrost, M. Mohseni and S. Lloyd, "Quantum Support Vector Machine for Big Data Classification", Vol -14, pp. 3-8, American Physical Society, 2014.
2. Vapnik. V. N., Levin E., Le Cun Y "Measuring the VC-dimension of a learning machine Neural Computation", 1994(6)
3. Osuna E. FreundR. Girosi F. "An improved training algorithm for support vector machines". Proc IEEE NNSP'97. Amelia Island FL,1997
4. John C.P. "Fast Training of Support Vector Machines using Sequential Minimal Optimization". In Scholkopf B. et al(ed.). Advances in Kernel Methods-Support Vector Learning, Cambridge, MA, MIT Press, 1999
5. Rong Xiao, Jicheng Wang, Fuyan Zhang, "An Approach to Incremental SVM Learning Algorithm". ICTAI, 2000(1): 2000
6. Jin-Long An, Zheng-Ou Wang, Zhen-Ping Ma "An Incremental Learning Algorithm for Support Vector Machine", IEEE, 2003.
7. P. Rebentrost, M. Mohseni and S. Lloyd, "Quantum Support Vector Machine for Big Data Classification", Vol -14, pp. 3-8,, American Physical Society, 2014

8. S. Lloyd, M. Mohseni and P. Rebentrost, "Quantum Algorithms for Supervised and Unsupervised Machine Learning", Quantum Physics, Springer, Vol-1, pp. 1-11, 2013.
9. IBMQ - https://www.research.ibm.com/ibm-q/
10. QISKIT - https://qiskit.org/

AUTHORS PROFILE

Dr.Balika.J.Chelliah (M.Tech, Ph.d.)-Associate Professor at CSE Department SRMIST, Ramapuram, Chennai. He is the acting guide for this paper and has keen interest in Artificial Intelligence and Software Engineering. He has publishednumerous papers in International Conferences and Journals.

Shristi Shreyasi - UG Scholar at CSE Department, SRMIST, Ramapuram, Chennai. She has keen interest in Machine Learning, Quantum Machine Learning.

Ananya Pandey– UG Scholar at CSE Department, SRMIST, Ramapuram, Chennai. She has keen interest in Image Recognition & Image Classification of ML.

Kirti Singh– UG Scholar at CSE Department, SRMIST, Ramapuram, Chennai. She has keen interest in IOT, Quantum Mechanics and Arduino.