Support Vector Machines: From Classical Version to Quantum

Gildas Decadjevi, Department of Computer Engineering, Ondokuz Mayıs University, 55139 Samsun, Turkey*
Sedat Akleylek, Department of Computer Engineering, Ondokuz Mayıs University, 55139 Samsun, Turkey
Gazmor Abdiu, Department of Computer Engineering, Ondokuz Mayıs University, 55139 Samsun, Turkey
Omar Halabi, Department of Computer Engineering, Ondokuz Mayıs University, 55139 Samsun, Turkey
**Corresponding author: gildasdeca@gmail.com*

Keywords: Support vector machine, machine learning, quantum computing, qubit
Discipline: Computer Science and Engineering

Abstract

Machine learning algorithms are used to analyze and process large amounts of data in order to get useful insights from them. Its combination with quantum computing gives rise to the field of quantum machine learning based on the implementation of machine learning algorithms on quantum computers in order to compare their performances and evaluate the possibilities to improve them. One of these algorithms is the support vector machine (SVM). This one is based on the determination of the optimum hyper plane that will separate the data in different classes. It has a time complexity of $O(M^2(M+N))$ on classical computers. However, its quantum version, the quantum SVM are theoretically proved to have logarithmic complexity. Through this work we carry out the SVM algorithm's study, its implementation on classical and quantum computers as well as a comparative analysis of the results. It shows that while the quantum SVM are faster than the classical SVM in matter of time, its accuracy still needs to be improved. For that purpose, we propose that the development and application of noise reduction methodologies will help getting better performances.

Introduction

Machine learning algorithms are used to build models for data analysis in various domains such as financial services, health care, sales, retail, transport, and much more. They are used in financial services for fraud detection, credit scoring, and stock trading [1]. In health care, machine learning has enhanced the processes of drug design, disease detection, and treatment effectiveness evaluation [2]. It's also improving the sales and retail sector through sales forecasting, data-driven decision making, recommendation systems, and customer relationship management [3], [4]. All these areas have in common the concept of data. With the continual growth of the amount of data available, making sense of it, analyzing it, drawing useful insights that will help in decision making becomes a global necessity [5]. Thus, we have different machine learning algorithms that are developed and continuously improved for this purpose and implemented on our classical computers [6]. Although there are many algorithms for a good analysis of the data, classical machine learning is facing different challenges such as the performance of its algorithms and their complexity: the processing performance, the curse of modularity, the curse of dimensionality, the class imbalance, feature engineering, and non-linearity [5]. The larger the amount of data to analyze is, the higher the required computing power is [7]. Classical devices do not always execute such tasks easily due to their limited capacities of computation and memory. It's then important to look for possible ways to increase the computation capabilities of those devices as well as trying to improve the existing machine learning algorithms in order to have lesser complexity.

Quantum computing's main promise to machine learning is to provide not only a higher capacity of computation on quantum computers but also a new way to apply those algorithms so as to get more efficient results [8]. Quantum physics makes it possible to encode and process information in ways that completely exceed what classical computers would actually be capable to do [9]. Quantum entanglement and teleportation are some non-classical phenomenons that give quantum computing an edge over classical computing [10]. With these features, quantum computers are expected to run quantum versions of many existing algorithms with an exponential speedup [11]. The study of machine learning algorithms with a quantum-based approach enabled the emerging of the new field of quantum machine learning (QML). QML is focused on the research and implementation of software that will enable the use of the advantages made available by quantum computing to enhance machine learning and its quest for efficiency and speed in a world of voluminous data [12]. In data analysis, one of the most known algorithms is the support vector machines (SVM). It is mainly used in classification or regression problems with multiple applications [13]. The quantum version of the SVM namely the quantum support vector machine (QSVM) is theoretically proved to have an exponential speed-up comparatively to the classical SVM [14]. It will be then a great revolution to get such a result practically by implementing successfully this widely used algorithm.

Motivation and Contribution

The field of quantum machine learning presents a lot of sub-areas to explore. Due to the fact that it is a new domain, the researches, as well as the implementations, are still at an early stage, making it possible to engage in one of its sub-areas for new discoveries or to work on ways to translate the theoretical knowledge available into practical applications that will enable its use in the industry. Different quantum machine learning algorithms have been developed but not implemented on the real quantum devices for reasons such as their unavailability or their inaccessibility. But now, with the different simulators and even some real devices made available online by some high-tech companies, the implementations of those algorithms have started. In the same direction, in this work, we apply classical SVM to classify a breast cancer data set and applied quantum SVM on the same data set but on IBM quantum devices namely Ibmq_melbourne, Ibmq_rome and Ibm_qasm simulator. After the analysis and implementation of the SVM and QSVM algorithm, we performed a comparison of their speed and accuracy which shows that the theoretical supremacy of the QSVM is not yet practically obtainable with the available devices due to the noise factor.

Organization

The rest of this paper is structured as follows. In Section II, we present the main background and explanation of the studied algorithm which is the SVM. In Section III, we explain the quantum SVM. In Section IV, the algorithm implementation, the data set, and the results have been presented followed by an analysis and suggestions for future results improvement. The conclusion of the work is provided in Section V.

Preliminaries

In this section, the SVM algorithm is clearly explained. Machine learning includes a large variety of algorithms that can be classed as supervised learning, unsupervised learning, and reinforcement learning algorithms [15]. In supervised learning, the data is labeled. Thus, the model basing on the expected output of the received input infers a function which' accuracy is measured during the testing phase. Contrarily to that, in unsupervised learning, the data is unlabeled and should be classified in groups based on some common characteristics or properties to be detected by the model. In reinforcement learning, the model receives the inputs and is directed or taught while producing an output for the different inputs basing on a rewarding system. SVM are among supervised learning algorithms. SVM's main use is to classify the data into the right groups while striving to find and establish the optimum hyper plane that will enable the categorization. The sought hyper plane's equation is given in Equation (1) [16].

$$w^T x \pm b = 0 \tag{1}$$

Accordingly, to the number of data groups, there are binary SVM and multi category SVM. Binary SVM are used when the data can be classified into two groups. In binary classification, the training set is of the type $\{(X_1, Y_1), (X_2, Y_2), \dots, (X_n, Y_n)\}$ with$Y \in \{-1, 1\}$ having X_i as input and Y_i as output. The data is said to be linearly separable when the two groups can be divided by a straight line and it is qualified as non-linear in the opposite case. In Figure 1, a representation of linearly separable data is given.

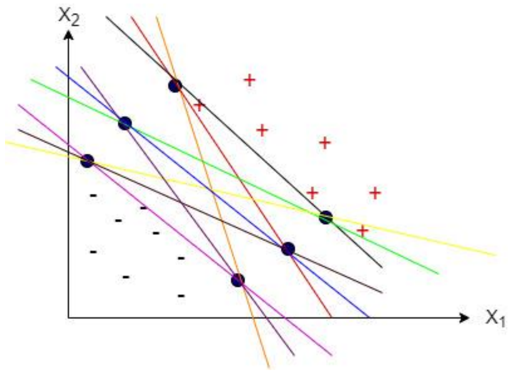

Figure 24. Linearly separable data

In this figure the goal is to classify the positive data points $(+1)$ from the negatives (-1). There are many possible lines that can divide them but the SVM looks for the hyper plane that will yield the maximum gap between the data points of the two classes. That distance is the margin and the model is named hard margin SVM. The data points present on the boundaries are the support vectors. The equations of the plans of the positive and the negative support vectors are respectively given in (2) and (3) [16].

$$w^T x - b = +1 \tag{2}$$

$$w^T x - b = -1 \tag{3}$$

w is the normal vector to the hyper plane, and b is the bias parameter helping to determine the offset of the hyper plane from the origin. When the plans are parallel, they have the same normal vector but their bias value is not the same. Thus, we can determine the gap between the positive and negative boundary plans as given in Equation (4).

$$\frac{|b_1 - b_2|}{||w||} = \frac{|1 - (-1)|}{||w||} = \frac{2}{||w||} \tag{4}$$

As the goal of the SVM model is to maximize the margin, then $\frac{2}{||w||}$ is to be maximized, which means that we will minimize $\frac{1}{2}||w||^2$. For the optimum classification of the positive and negative data points, it is expected that none of them will fall into the space delimited by the margin. This condition is expressed by the Equations (5), (6) and generalized in (7) [16].

$$w^T x - b \geq +1 \; with \; y = +1 \tag{5}$$

$$w^T x - b \leq -1 \; with \; y = -1 \tag{6}$$

$$y(w^T x - b) \geq 1, for \;\; x = 1, \ldots, N \tag{7}$$

These conditions lead to the primal formulation of hard margin SVM given in a reduced way in Equation (8) so that for positive data points $f(x) = +1$ and for negative ones $f(x) = -1$ [17].

$$f(x) = sign(wx - b) \tag{8}$$

For optimization purposes in linear SVM, we apply the method of Lagrange multipliers to the primal expression so as to obtain the dual formulation of hard margin SVM given in the Equations (9), (10), (11) and (12) [18].

$$\max_{\vec{\alpha}} L(\vec{\alpha}) \sum_{j=1}^{M} -\frac{1}{2} \sum_{j,k=1}^{M} \alpha_j x_j \alpha_k x_k \, , \sum_{j=1}^{M} a_j y_j = 0 \tag{9}$$

$$\vec{w} = \sum_{i=1}^{N} \alpha_i y_i \vec{x}_i \tag{10}$$

$$f(\vec{x}) = sign(\sum_{i=1}^{N} \alpha_i y_i \, \vec{x}_i \cdot \vec{x} + b) \tag{11}$$

$$b = \frac{1}{|\{i: \alpha_i \neq 0\}|} \sum_{i:\alpha_i \neq 0} (w^T x_i - y_i) \tag{12}$$

Due to factors such as noise in the data set, some data points may fall on the wrong side of the hyper plane creating confusion in the classification. In that case, we define a trade-off variable C slack variable representing the distance between the outliers and the hyper plane with the goal to minimize these distances. This forms the soft margin SVM model expressed through the Equations (13) and (14). In the same way, the primal formulation of soft margin linear SVM as well as the dual formulation are given in the Equations (15), (16) and (17), respectively [16].

$$w^T x_i + b \geq 1 - \xi_i \text{ for } y_i = +1 \tag{13}$$

$$w^T x_i + b \leq -1 + \xi_i \text{ for } y_i = -1 \tag{14}$$

$$\min \frac{1}{2} \sum_{i=1}^{n} w_i^2 + C \sum_{i=1}^{N} \xi_i \tag{15}$$

$$y_i(\vec{w} \cdot \vec{x}_i + b) \geq 1 - \xi_i \text{ , } i = 1, \dots, N \tag{16}$$

$$\begin{gathered} max \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{N} \alpha_i \alpha_j y_i y_j \vec{x}_i \cdot \vec{x}_j \\ 0 \leq \alpha_i \leq C \text{ and } \textstyle\sum_{i=1}^{N} \alpha_i y_i = 0,\ i = 1, \dots, N \end{gathered} \tag{17}$$

When the trade-off value C is very high, the soft margin SVM tends to be equivalent to the hard margin SVM. However, when C is very low, the probability of erroneous classification increases [16]. When the binary classification is to be done on a data set of which the points are not linearly separable, the SVM translates the data from its original space to a higher dimensional space named the feature space so as to determine the classification hyper plane in that space.

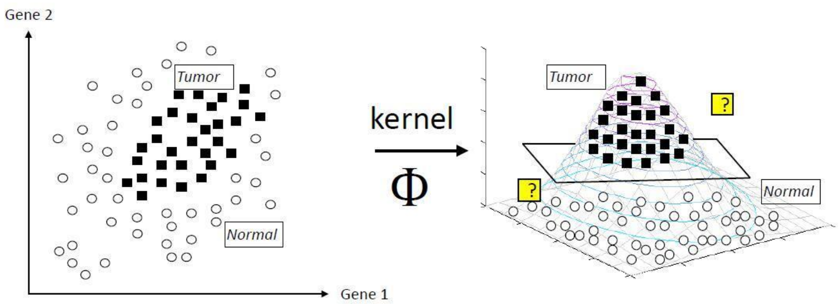

Figure 25. Classification with non-linear SVM

In Figure 2, an example is given showing the space translation and data classification by non-linear SVM [16]. The primal formulation of hard margin non-linear SVM is given in Equations (18), (19), and (20) as follows [16].

$$w \cdot \phi(x) + b = 0 \tag{18}$$

$$argmin_{w,b} \frac{1}{2} ||w||^2 \tag{19}$$

$$y_i(w \cdot \phi(x_i) + b) \geq 1, \tag{20}$$

$\phi(x)$ is the mapping function from the input space to the feature space. Equations (21) and (22) presents the state in input space while (23) and (24) present it in the feature space [16].

$$f(x) = sign(\vec{w} \cdot \vec{x} + b) \tag{21}$$

$$\vec{w} = \sum_{i=1}^{N} \alpha_i y_i \vec{x}_i \tag{22}$$

$$f(x) = sign(\vec{w} \cdot \phi(\vec{x}) + b) \tag{23}$$

$$\vec{w} = \sum_{i=1}^{N} \alpha_i y_i \phi(\vec{x}_i) \tag{24}$$

When we replace w in $f(x)$ we get the equations (25) and (26).

$$f(x) = sign(\sum_{i=1}^{N} \alpha_i y_i \phi(\vec{x}_i) \cdot \phi(\vec{x}) + b) \tag{25}$$

$$f(x) = sign(\sum_{i=1}^{N} \alpha_i y_i K(\vec{x}_i, \vec{x}) + b) \tag{26}$$

$$\phi : R^N \rightarrow H \text{ for } K : R^N \times R^N \rightarrow R$$

In data classification and regression analysis, another version of SVM are Least-squares support vector machines (LSSVM). They are summarized into the optimization problem expressed in Equation (28) under the constraints presented in Equation (29) [18].

$$\text{argmin } \frac{1}{2} w^T w + \frac{\gamma}{2} \sum_{i=1}^{M} e_i^2 \tag{28}$$

$$y_i(w^T \phi(x_i) + b) = 1 - e_i, \tag{29}$$

The parameter γ determines the cost or the trade-off value. By applying the method of Lagrange, we obtain the Equation (30) with K as kernel matrix [18].

$$\begin{pmatrix} 0 & 1^T \\ 1 & K + \gamma^{-1} I \end{pmatrix} \begin{pmatrix} b \\ \alpha \end{pmatrix} = \begin{pmatrix} 0 \\ y \end{pmatrix} \tag{30}$$

Apart from binary SVM, there are multi category SVM used when the data should be categorized into more than two classes. One of the main methods used is the one-versus-rest multi category SVM method. In this method, we determine the hyper plane corresponding to the classification of the element of only one class against all the other classes. In this case it is considered as if it were a binary classification where the i^{th} class is the first and all the $(n - 1)$ rest of class constitute second different class and the corresponding hyper plane is defined. After applying this to each of the n classes, we finally get n hyper planes which are analyzed with the decision functions so as to define the complete hyper plane of the multi category classification. The disadvantage of this method is related to all the computations that it requires.

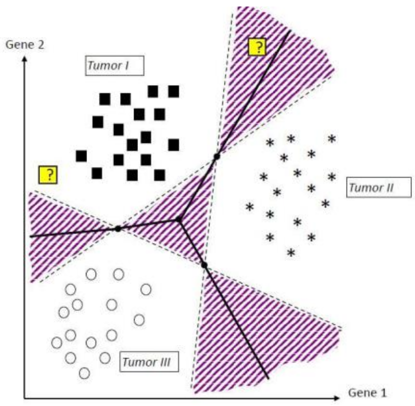

Figure 26. Example of one-versus-rest multi category SVM method.

In Figure 3, an example of multi category classification based on one-versus-rest is presented [16]. When this method is applied, the different equations explaining the functioning of the SVM model are presented in Equation (31), (32) and (33) [18].

$$\min_{w^i,b^i,\xi^i} \frac{1}{2}(w^i)^T w^i + C\sum_{j=1}^{N}\xi_j^i \tag{31}$$

$$(w^i)^T\phi(x_j) + b^i \geq 1 - \xi_j^i,\ y_i = i \tag{32}$$

$$\left(w^i\right)^T\phi(x_j) + b^i \leq -1 + \xi_j^i,\ y_i \neq i \tag{33}$$

Quantum Support Vector Machines

In this section, the quantum SVM are explained. As for classical SVM, quantum SVM are used in classification and regression processes. The main goal is also the determination of the optimum hyper plane. The difference between classical SVM and quantum SVM is how the algorithm is implemented leading then to differences of performance. Quantum SVM was developed primarily to solve the Least Square problem. The training set is expressed as a quantum state via the quantum random access memory (QRAM). Mathematically, the state's representation is given in Equation (34) [19]:

$$|\vec{x_j}\rangle = \frac{1}{|\vec{x_j}|}\sum_{k=1}^{N}(\vec{x}_j)_k\,|k\rangle \tag{34}$$

Quantum SVM can be realized using the Grover search algorithm when the data used is a classical one. The Grover search algorithm finds an element in an unordered set in $\mathcal{O}(\sqrt{n})$steps whereas classically at least $\mathcal{O}(n)$ steps are required. This added to the adiabatic quantum optimization leads to a quadratic speed up of the QSVM comparatively to classical SVM. If the data is already in quantum state or has been converted to quantum state, then the LS-SVM's quantum version can be executed on quantum devices. Thus the kernel matrix is calculated using quantum algorithm for inner product on QRAM, [20] and the other operations are performed using the quantum algorithm for solving linear equations. Finally, the trained qubits are used for the data classification with a logarithmic complexity $\mathcal{O}(log(MN))$. To get these results, the matrix problem that has been solved is expressed in Equation (35) with ways to find its inverse given in Equation (36) and (37) [18].

$$F\begin{pmatrix} b \\ \vec{\alpha} \end{pmatrix} = \begin{pmatrix} 0 & \vec{1}^T \\ \vec{1}^T & K + \gamma^{-1}I \end{pmatrix}\begin{pmatrix} b \\ \vec{\alpha} \end{pmatrix} = \begin{pmatrix} 0 \\ \vec{y} \end{pmatrix} \tag{35}$$

$$F = \begin{pmatrix} 0 & \vec{1}^T \\ \vec{1}^T & K + \gamma^{-1} I \end{pmatrix} \tag{36}$$
$$F = J + K\gamma \text{ such that}$$

$$J = \begin{pmatrix} 0 & 1^T \\ 1 & 0 \end{pmatrix} \text{ and } K\gamma = \begin{pmatrix} 0 & 0 \\ 0 & K + \gamma^{-1} I \end{pmatrix} \tag{37}$$

When the different quantum matrix algorithms and Lie algebra are applied, the normalization process of K and F matrices are performed and the inverse of F matrix is found. If it were the classic SVM, the complexity would be $O\big(M^2(M+N)\big)$, however, with this method of QSVM an exponential speedup is noticed [18]. In algorithm 1, the pseudocode of QSVM algorithm is presented [21].

Algorithm 1 Quantum SVM
Input:
- Training data set $(\vec{x}_j, y_j)$, $j = 1, 2, 3, \dots, M$
- A query data $\vec{x}$

Output:
- Classification of $\vec{x}$: $+1$ or -1

Procedure:
- Calculate kernel matrix $K_{ij} = \vec{x}_i \cdot \vec{x}_j$ using quantum inner product algorithm.
- Solve the linear equations and find $|b\vec{\alpha}\rangle$ using a quantum algorithm for solving linear equations (training step)
- Perform the classification of the query data $\vec{x}$ against the training results $|b\vec{\alpha}\rangle$ using a quantum algorithm.

In Table 5, a general comparison of classical SVM and quantum SVM is given [18] [17].

Table 5. General comparison of classical and quantum SVM

Comparison	**Classical SVM**	**Quantum SVM**
Learning type	Supervised learning	Supervised learning
Intended use	Classification and regression	Classification and regression
Hyper plane equation	$\vec{w}\vec{x} + b = 0$	$\vec{w}\vec{x} + b = 0$
Complexity	$O\big(M^2(M+N)\big)$, with M the number of training instances and N the number of dimensions in the feature space.	$O(M^2N)$, with Grover search algorithm $O(log(MN))$ with quantum matrix Optimization algorithms.
Algorithms that are used in the background	Standard matrix multiplication, linear equation solving and matrix inversion method.	Grover search algorithm, quantum linear equation solving algorithm, quantum sparse matrix method, PCA.

Experimental Results

In this section, the classical and the quantum version of SVM algorithm have been implemented, compared and analyzed. The two versions of the algorithm have been executed on the breast cancer Wisconsin diagnostic data set. It is available in the Scikit-learn library and has 569 instances with 30 numeric predictive attributes and the class. A cell can be classified either as benign or malignant. The data set has been initially preprocessed and normalized so as to ease the classification process. In each case, 70% of the data set constituted the training set and 30% the test set. On the classical device, the average result is given in the Figure 4.

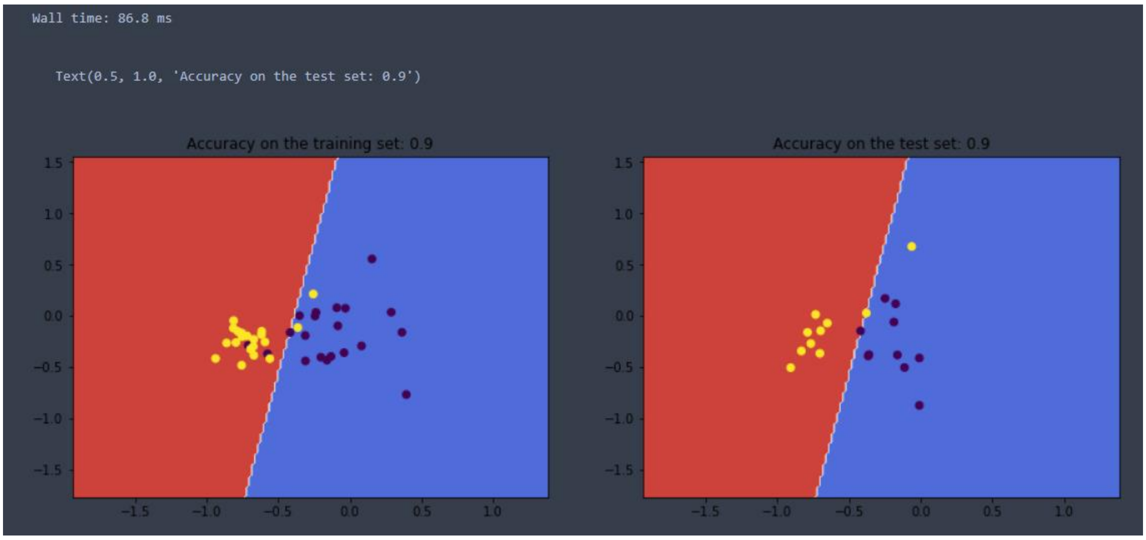

Figure 27. Data classification with classical SVM

For the quantum SVM, the dimensionality of the data has been reduced using the principal component analysis (PCA) before its classification.

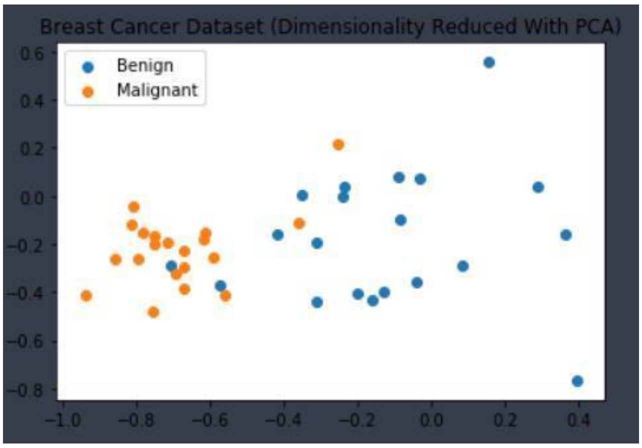

Figure 28. Data set with dimensionality reduced with PCA

In Figure 5, the representation of the data after dimensionality reduction with PCA is given. The quantum SVM has been run on the IBM Qiskit simulator as well as two of the real quantum devices made available by IBM and reachable online for experimental purposes. In Table 2, the comparison of the performance results of the classical SVM and quantum SVM is provided. In Figure 6, the circuit of the representation of QSVM on a 5-qubits real device is presented [22].

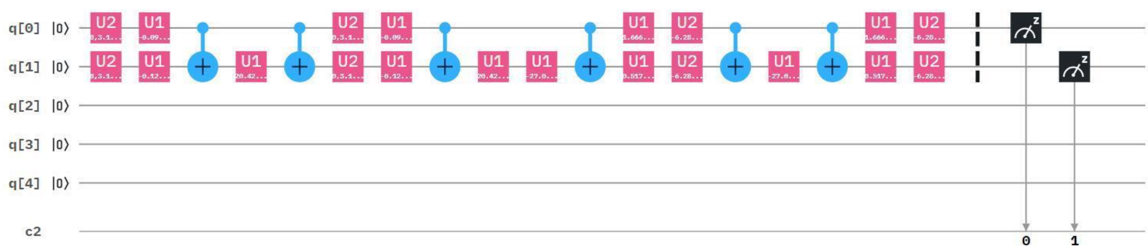

Figure 29. Representation of QSVM circuit

Table 6. The performance differences based on the type of the SVM algorithm and the implementation device

Comparison	**Average Time**	**Average Accuracy**
Classical SVM on classical computer	75.8ms	90%
QSVM on local qasm_simulator	Seconds	95%
Quantum SVM on ibmq_16_melbourne (16 qubits)	50ms	70%
Quantum SVM on ibmq_rome (5 qubits)	50ms	70%

For instance, quantum SVM's execution on the two real quantum devices with different numbers of qubit results in average into the same performance in matter of time and accuracy. This means that the number of qubits unused does not influence considerably the result yielded by those in use. The fact that the classification time on the real quantum devices is lower than the classical one shows that the quantum computers are really faster than classical devices at computing tasks and that quantum SVM too are faster than classical SVM. The fact that the local simulator "qasm" has an average time that is higher than all the others proves how hard and costly it is to implement and test quantum properties on classical devices. The qasm-simulator is available offline to run directly on classical devices while approximating an ideal quantum environment (void of noise).

As shown in the table, the qasm simulator presents the highest accuracy among all the implementations. Given that the qasm simulator represents the ideal quantum computer environment, then it means that the real quantum devices used have not been able to work up to their full potential. It's also to be noticed that although those real quantum computers are faster than classical computers, they are not yet as accurate as classical computers are. In fact, if the quantum computers were working in ideal mode, they would produce at least the result of the local simulator or even better due to the limitations of the simulator on classical devices. Thus, one of the main reasons that could be hindering us from getting the best out of the quantum SVM expected to produce practically the exponential speedup, is the noise factor. In fact, quantum computers are designed to be held at a temperature that is completely close to the absolute zero which is not easy to achieve. For this reason, every modification or derangement of the environment may be an eventual source of noise, causing some errors to happen in the quantum information processing and leading consequently to less accuracy. Toward this situation, the suggestion of our work would be the exploration of possible ways of noise mitigation. Actually, one of the libraries being developed for quantum devices is the "Ignis" library that is an IBM framework for characterizing and mitigating noise in quantum devices. The study and application of this library to quantum SVM would then surely help to enhance the performances obtained actually on quantum devices for QSVM execution.

Conclusion and Future Works

In this paper, we carry out a large study of the quantum SVM, a QML algorithm starting from its main concepts as classical SVM until its implementation and the performance analysis. We find out that the theoretically proved exponential speedup of the quantum SVM comparatively to the classical SVM still requires some improvements before being achievable in practice. One of those requirements is the noise mitigation due to the fact that quantum devices are exposed to this factor that may corrupt their execution process and their results. For this reason, as future work, it is possible and preferable look for ways to minimize it by taking advantage of existing noise mitigation libraries or by building more efficient ones so as to be able to get the best out of these computers that have the capacity to revolutionize completely the way we process data.

References

[1] P. Domingos, "A few useful things to know about machine learning," Commun. ACM, vol. 55, p. 78–87, Oct. 2012.

[2] M. Haddadi and M. Dehghanian, "Strictly proximinal in normed spaces," World Applied Sciences Journal, vol. 16, no. 2, pp. 250–252, 2012.

[3] M. Bohanec, M. K. Borštnar, and M. Robnik-Šikonja, "Explaining machine learning models in sales predictions," Expert Systems with Applications, vol. 71, pp. 416–428, 2017.
[4] I.-F. Chen and C.-J. Lu, "Sales forecasting by combining clustering and machine-learning techniques for computer retailing," Neural Computing and Applications, 02 2016.
[5] A. L'heureux, K. Grolinger, H. F. Elyamany, and M. A. Capretz, "Machine learning with big data: Challenges and approaches," IEEE Access, vol. 5, pp. 7776–7797, 2017
[6] Y. Liu, J.-W. Bi, and Z.-P. Fan, "Multi-class sentiment classification: The experimental comparisons of feature selection and machine learning algorithms," Expert Systems with Applications, vol. 80, pp. 323 - 339, 2017.
[7] A. Katal, M. Wazid, and R. H. Goudar, "Big data: issues, challenges, tools and good practices," in 2013 Sixth international conference on contemporary computing (IC3), pp. 404–409, IEEE, 2013.
[8] T. A. Shaikh and R. Ali, "Quantum computing in big data analytics: A survey," in 2016 IEEE International Conference on Computer and Information Technology (CIT), pp. 112–115, IEEE, 2016.
[9] A. Galindo and M. A. Martin-Delgado, "Information and computation: Classical and quantum aspects," Reviews of Modern Physics, vol. 74, no. 2, p. 347, 2002.
[10] R. Jozsa, "Entanglement and quantum computation," arXiv preprint quantph/9707034, 1997.
[11] J. Biamonte, P. Wittek, N. Pancotti, P. Rebentrost, N. Wiebe, and S. Lloyd, "Quantum machine learning," Nature, vol. 549, no. 7671, pp. 195–202, 2017.
[12] W. S. Noble, "What is a support vector machine?" Nature biotechno-logy, vol. 24, no. 12, pp. 1565–1567, 2006.
[13] P. Rebentrost, M. Mohseni, and S. Lloyd, "Quantum support vector machine for big data classification," Physical review letters, vol. 113, no. 13, p. 130503, 2014.
[14] S. Shalev-Shwartz and S. Ben-David, Understanding Machine Learning: From Theory to Algorithms. USA: Cambridge University Press, 2014.
[15] R. Jozsa and N. Linden, "On the role of entanglement in quantum computational speed-up," Proceedings of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences, vol. 459, no. 2036, pp. 2011–2032, 2003.
[16] A. Statnikov, A gentle introduction to support vector machines in biomedicine: Theory and methods, vol. 1. world scientific, 2011.
[17] E. R. Johnson, N. Harrigan, and M. Gimeno-Segovia, Programming Quantum Computers: Essential Algorithms and Code Samples. O'Reilly Media, Incorporated, 2019.
[18] P. Wittek, Quantum Machine Learning: What Quantum Computing Means to Data Mining. 08 2014.
[19] Y. Huang, H. Lei, and X. Li, "A survey on quantum machine learning," Chin. J. Comput, vol. 41, no. 1, pp. 145–163, 2018.
[20] S. Lloyd, M. Mohseni, and P. Rebentrost, "Quantum algorithms for supervised and unsupervised machine learning," arXiv preprint arXiv:1307.0411, 2013.
[21] P. Coles and S. Eidenbenz et al., "Quantum algorithm implementations for beginners," 04 2018.
[22] V. Silva, "Enter the ibm q experience: A one-of-a-kind platform for quantum computing in the cloud," in Practical Quantum Computing for Developers, pp. 77–141, Springer, 2018.