Quantum Accelerated Support Vector Machines

Vincent van Engers[1] and Noor Mansour[2]

[1]University of Twente, MSc Computer Science, s2065630
[2]University of Twente, MSc Computer Science, s2206641

Quantum computing promises the acceleration of machine learning algorithms and the solving of classically intractable problems. The parallelism of quantum embedding and feature mapping has inspired research in quantum kernels to enhance kernel driven classical machine learning methods. The promises of quantum computing and its mystique created by popular culture obfuscates its capabilities and commercial applicability. This paper details the relation between quantum embedding and feature mapping and provides a empirical comparison of kernel driven algorithms: classical support vector machines (SVM), quantum support vector machines (QSVM) and quantum support vector classifiers (QSVC). Furthermore, it addresses the framing of quantum computing by discussing its current capabilities and challenges

1 Introduction

Since recent years, research has started focusing on the acceleration of machine learning by applying techniques from quantum computing to support vector machines (SVM). SVMs are linear classifiers that construct a hyperplane in high- or infinite-dimensional space that can be used for data separation. It constructs the hyperplane such that it maximizes the distances to the nearest data points. In order to analyze non-linear separable data, the SVM uses kernels to project data into higher dimensional Hilbert space, achieving linear separability. Kernel theory is closely related to data embedding in quantum computing and this relation is the basis of quantum support vector machines (QSVM). At the intersection of multiple fields, QSVMs are being approached from different directions. Rebentrost et al. (2014) [1] show how a regular SVM can be quantitized and run as a quantum algorithm. Chatterjee et al. (2017) [2] note the similarities between kernel theory and quantum mechanics from a quantum mechanical perspective. Stoudenmire et al. (2017) [3] show from a classical machine learning perspective how data encoded into quantum systems can be interpreted as feature maps. Schuld et al. (2019) [4] and Havlicek et al. (2019) [5] concurrently introduced QSVMs from a quantum computational approach.
This introductory paper explains QSVMs by clarifying the parallels between quantum embedding and kernel feature mapping. It is intended for audiences from a machine learning background with elementary knowledge of quantum computing. From the perspective of quantum kernels, two approaches to QSVMs are discussed: quantum kernel estimation (QKE) [5] and covariant quantum kernels with quantum kernel alignment (QKA) [6]. Both approaches will be a hybrid-quantum-classical approach with a quantum kernel and classical SVM. The theoretical explanation is followed by a practical implementation. Following is an address of the hype and overselling of quantum computing through a discussion of the practical uses of quantum devices in general and QSVMs in particular.[1]

2 Background

2.1 Classical kernel theory

Support vector machines are linear classifiers that use kernels to transform M-dimensional, linearly inseparable input data into higher dimensional Hilbert space $\mathcal{H}$ to achieve linear separability. Formally we can define $\phi : \mathcal{X} \mapsto \mathcal{H}$ as a feature

[1]The discussion in the Discussion section on the current capabilities, restrictions and challenges of quantum computing is the piece on "the hype and overselling of quantum computing" that is a replacement for homework assignment 4.

map that maps an input $x \in \mathcal{X}$ to a Hilbert space $\mathcal{H}$, called the feature space. The inner product of two feature vectors can then be defined as a kernel: $K(x, x') = \langle \phi(x), \phi(x') \rangle$. Each kernel function is based on calculating the inner product of two vectors. Consequently each mapping into higher dimensional space through some function $\phi : \mathcal{X} \mapsto \mathcal{H}$ can be achieved without explicitly computing ϕ [7].
In kernel density estimation, the ground truth is estimated, by modeling the density of the probability distribution. This is achieved by anchoring similarity measures such as Gaussian kernels at each data point and adding all kernels up to achieve a density function. The kernel type and its parameters such as the Gaussian height and width might be adjusted to find a projection appropriate for linear separation. To perform classification, exclusively kernels anchored on same-classed data points are summed to achieve class-based density functions. New data points are then classified into the class with the highest density at that point:

$$\tilde{y} = \sigma \left(\sum_{m=1}^{M} y^m k(x^m, \tilde{x}) \right) \tag{1}$$

where $\tilde{y}$ is the predicted class, y is the label, k is the kernel and σ the activation function.

2.2 Quantum enhanced kernels

In quantum computing the inner product of the feature vectors can be viewed as two overlapping quantum states $|\phi(x)\rangle, |\phi(x')\rangle$: $\langle \phi(x) | \phi(x') \rangle$. The quantum state $|\phi(x)\rangle$ can be prepared through a state preparation routine as a dependent of some input S_x such that we have $|\phi(x)\rangle = S_x|0...0\rangle$. This quantum state can thus be interpreted as a feature vector in Hilbert space. As such the mapping of input to Hilbert space in classical kernels is analogous to quantum embedding. The inner product or overlap of two quantum states that are prepared in the same manner and with dependence on different input is a quantum kernel.
Dependent on the physical hardware, there are different ways in which the inner product can be efficiently computed and how the data points are loaded into the quantum device. Figure 1 shows a simple quantum circuit that computes the inner product. The first qubit is on an ancilla register. The other qubits encode the quantum feature vector. Figure 1 shows two of such qubits, however these can be extended depending on the input space. After the first Haddermat gate, the first qubit is in uniform superposition. S_x embeds the input into a quantum state and is conditioned on the ancilla qubit being in state $|1\rangle$. Any procedure that encodes input into a quantum state is a kernel. S_x takes form as one of a variety of embeddings such basis embedding. After S_x, the quantum feature vector, that encodes the input x, is entangled with the ancilla qubit being in state $|1\rangle$. $S_{x'}$ is similar to S_x, but with input x' being conditioned on the ancilla being $|0\rangle$. After $S_{x'}$, the two feature vectors are in one quantum superposition flagged by the sate of the ancilla qubit: $\frac{1}{\sqrt{2}}(|0\rangle)\phi(x')\rangle + |1\rangle\phi(x)\rangle)$. After applying the final Haddermat gate on the ancilla, measuring the ancilla at 0 has probability $p(0) = \frac{1}{2} + \frac{1}{2}\text{Real}\{\langle \phi(x) | \phi(x') \rangle\}$. In machine learning, kernels are interpreted as similarity measures wherefore the kernel will generally already be real-valued. In this configuration, the quantum circuit device can be used to create a hybrid quantum-classical algorithm where the kernel computation is outsourced to the quantum circuit and whose output is fed into the classical classifier such as a support vector machine.

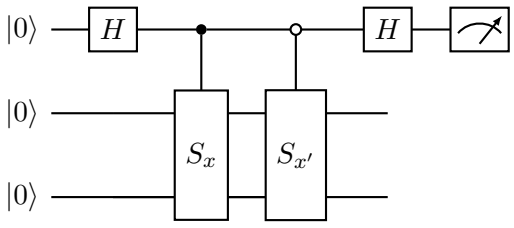


Figure 1: A simple quantum circuit where S_x and S_x respectively embed x and x' into a single quantum state.

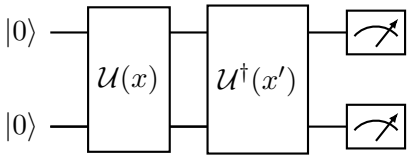


Figure 2: A quantum circuit depicting the Quantum Kernel Estimation (QKE) approach that utilizes the fact if $\mathcal{U}(x) = \mathcal{U}(x')$, the input should equal the output.

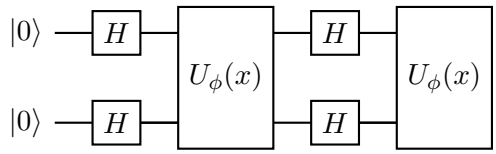


Figure 3: Decomposed $\mathcal{U}(x)$ feature map from figure 2 consisting of Hardamad gates and diagonal phase gate components $U_{\Phi(\vec{x})}$

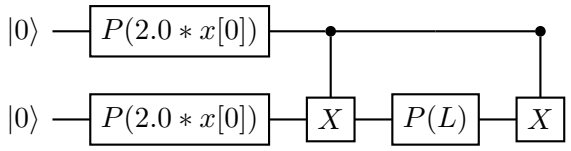


Figure 4: Decomposed subcomponent $U_{\phi(x)}$ of the ZZ feature map, where $L = 2.0 * (\pi - x[0] * [\pi - x[1])$

Instead of exclusively using the quantum circuit to provide kernel calculations, the data can be analyzed directly in feature Hilbert space. The state preperation routine S_x can be used to embed the data into a quantum state. Instead of subsequently loading x' and calculating the inner product, a variational circuit U_θ can be subsequently added, where θ is being optimized. The probability of the measured qubit is interpreted as the class prediction of the quantum classifier. Good kernels aid in data analysis or pattern recognition and should be able to achieve acceleration on quantum devices compared to classical computing devices. When the kernel of the quantum circuit in Figure 1 is too simple, such as a kernel feature map that exclusively calculates products, it provides no practical speedup compared to classical computation [5]. In addition to handcrafting quantum circuits, good kernels can be achieved by parameterization through variational circuits. The embedding S_x can be parameterized into $S_x(\phi)$ such that the kernel is not static but can be learned.

3 Empirical Investigation

3.1 Methodology

We compare classical SVM, QSVM that utilizes Quantum Kernel Estimation (QKE) and Quantum Support Vector Classifier (QSVC) that utilizes the Quantum Kernel Alignment (QKA) technique. The quantum-kernel methods are simulated on a classical computer with a Radial Basis Function (RBF) Kernel. We evaluate the performance of these three models on six different data-sets, half of which are binary classification tasks and the other half are multi-class classification tasks.

The QSVM follows the quantum kernel estimation approach of Havlicek et al. (2019) [5]. The kernel is calculated on a simulated quantum computer and used to transform the input data for classification in a classical SVM. The results of the QSVM are then compared to results of a purely classical SVM without the quantum kernel. A quantum kernel can be described as

$$K^Q(x, x') = |\langle \phi(x), \phi(x') \rangle|^2 \tag{2}$$

Given an encoding circuit $\mathcal{U}(x)$, this can be further formulated as

$$|\langle \phi(x), \phi(x') \rangle|^2 = |\langle 0^{\otimes N} | \mathcal{U}(x) \mathcal{U}^\dagger(x') | 0^{\otimes N} \rangle|^2 \tag{3}$$

By measuring the probability of observing $|0^{\otimes n}\rangle$ at the output after running the circuit as shown in figure 2, we can estimate the quantum kernel function as described in Eq. 3. It utilizes the fact that if $\mathcal{U}(x) = \mathcal{U}(x')$, the unitary in figure 2 and Eq. 3 would simplify to 1, resulting the input to equal the output. Then measuring the frequency of the $|0^{\otimes n}\rangle$ output , yields an estimation of the overlap of the two states as defined by the kernel in Eq. 2. This approach of estimating the quantum kernel is called Quantum Kernel Estimation (QKE).

Based on Havlicek et al. (2019) [5], Figure 3 shows the decomposition of a possible feature map $\mathcal{U}(x)$. Here, we first apply Hadamard gates to superpose the input, followed by a diagonal phase gate component $U_{\Phi(\vec{x})}$.[2] This is repeated to achieve a depth of 2, resulting in the following definition of the feature map circuit:

$$|\phi(x)\rangle = U_{\Phi(\vec{x})} H^{\otimes n} U_{\Phi(\vec{x})} H^{\otimes n} |0\rangle^{\otimes n}$$

The anticipated advantages of the QSVM might be obtained if a quantum feature map is chosen that is difficult or unfeasible to obtain with a classical computer [5]. Figure 4 shows the first part of the decomposed $U_{\Phi(\vec{x})}$ circuit.

[2]Beware that $\mathcal{U}(x) \neq U_{\Phi(\vec{x})}$

This first part is just repeated again in the case of two repetitions to obtain the full circuit. The example circuit on two qubits has parameterized phase gates, followed by a control x gate, which is then followed by another parameterized phase gate and concluded with another control x gate. This feature map has linear entanglement, which means that qubit i is entangled with qubit $i+1$ for all $i \in \{0, 1, \ldots, n-2\}$, for n total qubits. $U_{\Phi(\vec{x})}$ can be formally written as

$$U_{\Phi(\vec{x})} = \exp\left(i \sum_{S \subseteq [n]} \phi_S(\vec{x}) \prod_{i \in S} Z_i\right) \quad (4)$$

The circuit shown in Figure 3 in conjunction with the decomposition of $U_{\Phi(\vec{x})}$ in figure 4 define the so called second-order Pauli-Z evolution circuit.

The QSVC utilizes QKA to iteratively train towards a good quantum kernel for a given problem. SVMs require a choice for the specific kernel function. In practice, the symmetries of the dataset are consulted to inform the decision for a kernel or the decision is made arbitrarily. A kernel chosen based on the structure of the data is likely to generalize better [6]. Therefore Glick et al. (2021) [6] proposed a Quantum Kernel Alignment (QKA) algorithm that iteratively adapts the kernel to have high similarity to a target kernel that is informed based on the structure of the underlying data distribution. The resulting kernel that is aligned on the data has better generalization capabilities to unseen data than unaligned kernels because it exploits group structures in the data. The QKA algorithm aims to maximize the alignment of the kernel while converging to the maximum SVM margin by optimizing parameters in a quantum circuit. These so called covariant quantum kernels are defined as follows:

$$K_\lambda(x, x') = \left|\left\langle 0^n \left| V_\lambda^\dagger D^\dagger(x) D(x') V_\lambda \right| 0^n \right\rangle\right|^2 \quad (5)$$

The feature map here is defined as the unitary $D(x)$ of a group G for some element $x \in G$. QKA is then used to find a good fiducial reference state $V_\lambda|0^n\rangle$, which is parameterized by λ, for a given group. For our specific experiments, the quantum circuit of our feature map is showcased in figure 5. The trainable rotational layer give us the opportunity to optimize the parameter λ over the loss function with the Simultaneous

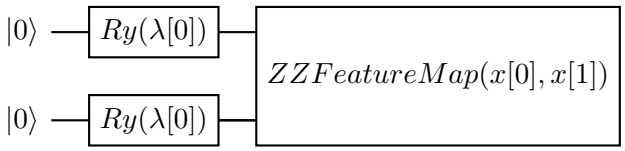


Figure 5: Trainable rotation layer with ZZFeatureMap from figure 3 and figure 4.

Perturbation Stochastic Approximation (SPSA) optimizer. The SVC loss function is defined as:

$$loss = \sum_i a_i - 0.5 \sum_{i,j} a_i a_j y_i y_j K_\lambda(x, x') \quad (6)$$

x_i are the training samples with labels y_i and a kernel K_λ, parameterized by λ, where a_i are the optimal Lagrange multipliers found by solving the standard SVM quadratic program. By minimizing this loss over the parameter λ, the weighted kernel alignment is maximized, which yields the smallest upper bound on the generalization error of a given parameterization [6].

Three of the six datasets contain more than 2 classes and therefore need some special attention. In case of a multi-class classification, there are 2 options that define how the SVM approaches this task: all-pairs and one-against-rest. For a n-way multi-class classification problem, the all-pair method trains $n(n-1)/2$ binary SVM's that each get a pair of classes to train on. So each binary SVM learns how to distinguish the two classes it got and at prediction time, a weighted voting scheme is used. The weighted voting scheme applies all $n(n-1)/2$ binary SVM's to the test data point and sums up the scores from the SVM's for each class. Then the class with the highest value is chosen as the prediction. The one-against-rest method creates n SVM classifiers, where the i-th classifier trains to classify class i from all other classes $\forall i \in 1, 2, \ldots, n$. Then at prediction time, the classification is based upon the classifier that predicts the highest value for that unseen data-point and returns the corresponding class label. For our experiments the one-against-rest method is used to approach the multi-class classifications due to its computational efficiency.

3.2 Results

Table 1 shows the test accuracy's of our experiments, in which we tested the classical SVM

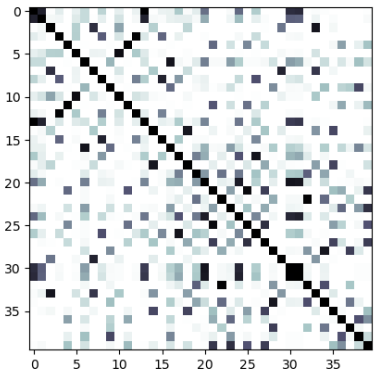
Dataset	Classes	SVM	QSVM	QSVC
ad_hoc	2	0.65	**1.00**	0.80
breast_cancer	2	0.85	**0.90**	0.70
gaussians	2	0.55	0.50	**0.60**
wine	3	**1.00**	0.90	0.93
iris	3	1.00	1.00	1.00
digits	9	**0.60**	0.39	0.55

Table 1: Testing Accuracy of classical SVM vs. QSVM (QKE) vs. QSVC (QKA) on 6 data-sets with different number of classes to classify.

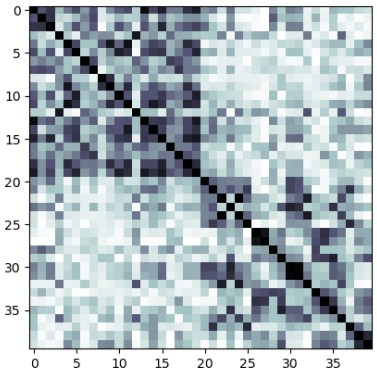
approach against its quantum counterpart on 6 data-sets. In general we can observe that the quantum approaches to SVM have higher accuracy for the binary classification tasks than the classical SVM, but the SVM performs better for multi-class classifications. For the ad-hoc and breast-cancer data, the QSVM performs better than the classical and QSVC. The QSVC performs worse than the QSVM but better than the classical SVM. The QSVC also performs worse than the QSVM and the SVM for the breast-cancer data. For the binary classification of the Gaussian data, the classical SVM performs better than the QSVM, while the QSVC has the highest accuracy at 60%.

Regarding the multi-class classifications of the wine (3 classes), iris (3 classes) and digits (9 classes) dataset, the classical SVM achieves the highest accuracy in the wine and digits datasets. For the iris dataset, SVM, QSVM and QSVC match each other in performance. Especially the difference in the accuracy of the digits dataset, shows the better performing classical SVM. The QSVC is more accurate in all of the multi-class problems than the QSVM, while still being slightly worse than the SVM. In all cases where the SVM is better than QSVM, the QSVC performs better than the QSVM.

Figure 6 presents the kernels computed by the classical SVM (6a) and the kernel computed in the quantum space (6b. The elements of the matrix correspond to the similarities between data points in the input space [8]. The kernel computed classically has sporadic similarities, with some regular similarity pattern between data points of 30 and others. Contrasting, the quantum kernel has two larger groups of data samples with high similarity.



(a) Classical SVM Kernel on ad hoc data



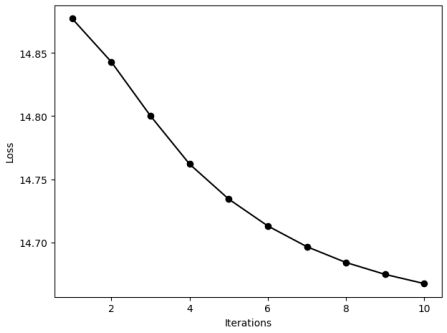
(b) Quantum Kernel on ad hoc data

Figure 6: Kernel of SVM vs QSVM

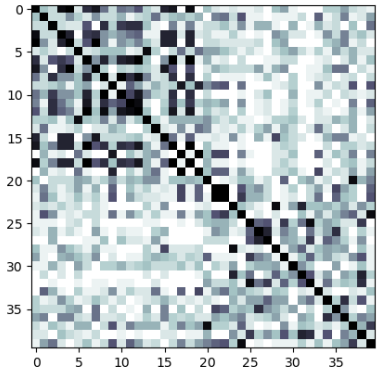
Figure 7 show the value of the loss function over several iterations during the optimization procedure (7a) while also presenting the kernel generated with the QKA approach(7a). The kernel resembles the Kernel from 6b in that it also has two groups of data samples with high similarity, although the similarity is not as dense and is less established. The decrease of the loss value seems to follow a near exponential decay and then starting to converge to a value at the end.

4 Discussion

The combination of classical machine learning algorithms and quantum computing principles promises the acceleration of computation time in machine learning, the increase of prediction performance and the ability to compute intractable problems. The application of quantum principles in machine learning is largely limited by the contemporary state-of-the-art quantum computing hardware. Quantum computers are subject to fragility and noise, leading to computational errors [9]. Error correction has been long studied in



(a) Loss Value over Iterations



(b) Quantum Kernel after QKA optimization

Figure 7: Iteration and Kernel of QKA approach on ad-hoc data

classical information theory [10] and many quantum quantum codes and techniques have been introduced in the form of quantum error correction [11]. Nonetheless, error correction increases the required qubit count for computation. Noiseless computation of problems that are practically useful far exceed the qubit count of contemporary systems [12, 13]. There are initiatives to bypass the requirement of quantum error correction by using qubits that are subject to less instability in topological quantum computing [14, 15]. Yet such approaches face their own hardware challenges. Contemporary quantum computing devices are denoted as near-future quantum devices or noisy intermediate scale quantum (NISQ) [9]. The premise is that small noisy collections of qubits could outperform classical computers on some metric and prove practically useful. NISQ is hailed by quantum startups, but commercial value remains unclear [12]. A recent experiment using 20 noisy qubits has shown the previously theoretically predicted aspect of time crystals in quantum computing [16]. Yet the experiments uncovered no new understanding of time crystals nor computational benefits compared to classical computers which can already simulate such time crystals. Similar NISQ experiments simulate random quantum circuits [17]. The research is heralded in the context of exponential acceleration and quantum supremacy; practically, it computes a highly specialized task with no economic or commercial value [12]. Realistically, NISQ can be viewed as a testbed to explore the possibilities at the intersection of quantum computing and machine learning, while quantum hardware is being developed beyond the contemporary minimal qubit counts. QSVMs fall into the category of NISQ devices. As such, researchers do not present QSVMs as replacements of classical SVMs but as preliminary investigations into the potential of quantum machine learning [5, 4]. Still, SVMs suffer from inherent problems that could be addressed by QSVMs. First classical kernels suffer from polynomial scaling where computations typically scle at the order of n^3 with n data points, limiting the applicability of kernel-based methods and SVMs [18]. Second, due to the kernels corresponds to the nonlinear mapping of the input data, it is essential to use the appropriate kernel for a given dataset [18]. QSVMs might increase the usability of kernel-based methods being of logarithmic complexity instead of polynomial complexity [1]. Quantum kernels can generate feature maps that are difficult for classical kernels to compute, however they still suffer from great dependency on datasets [18]. Furthermore, quantum kernels and quantum learning in general tend towards overfitting and require special regularization to produce representative results. As shown in Section 3.2, prediction performance can drop significantly if there is a misfit between dataset and kernel [19]. QSVM's do not always perform better than their classical counterpart, as our experiments empirically prove. QSVM's mostly obtain a quantum advantage for data that needs a kernel which would not be feasible in classical computing or computationally challenging to obtain. In the cases where the SVM performed better than the QSVM, the QSVC trained with QKA performed better than the QSVM and approached or even surpassed the accuracy of the SVM. Training the kernel to incorporate the underlying data structure via the QKA approach can lead to increased performance and higher generalization capabilities of quantum kernel methods, underlying the need for a good

fit between kernel and dataset as one of the more impactful factors on accuracy.
Beyond the limitations of QSVMs, even in the role of testing environment for exploratory research, NISQ computing devices are limited in providing experiments as the aforementioned external noise and internal stability not only reduce available qubits in noiseless devices, but also limits its experimental run time [9]. As such the errors can accumulate quickly and render any results useless.
Overall, the benefits of quantum computing such as acceleration and solving of intractable problems are possible, but their realisation in practical use cases is still potential rather than manifest. Quantum computing is still in an exploratory phase, limited by contemporary state-of-the-art quantum hardware. Current quantum devices are prone to erroneous predictions due to external noise and internal instability. The NISQ definition is a re-framing of current quantum devices rather than a technical leap. It is a method to exploit the current generation of quantum devices to its fullest extend, rather than a leap toward a next generation. Similarly, the current state of quantum kernel-devices are means to explore the possibilities and drive innovations in quantum machine learning.

5 Conclusion

This paper shows how the parallelism between quantum embedding and classical feature mapping inspire the development of quantum-kernel-driven machine learning. We compare SVMs, QSVMs and QSVCs and observe the same dependency between dataset and kernel as previous research has shown. Quantum computing might lead to great computational benefits, yet the current state of quantum computing is limited by the development of quantum devices and provides little commercial value.

References

[1] P. Rebentrost, M. Mohseni, and S. Lloyd, "Quantum support vector machine for big data classification," vol. 113, no. 13, p. 130503.

[2] R. Chatterjee and T. Yu, "Generalized coherent states, reproducing kernels, and quantum support vector machines," vol. 17, p. 1292.

[3] E. M. Stoudenmire and D. J. Schwab, "Supervised learning with quantum-inspired tensor networks." Number: arXiv:1605.05775.

[4] M. Schuld and N. Killoran, "Quantum machine learning in feature hilbert spaces," vol. 122, no. 4, p. 040504.

[5] V. Havlicek, A. D. Córcoles, K. Temme, A. W. Harrow, A. Kandala, J. M. Chow, and J. M. Gambetta, "Supervised learning with quantum enhanced feature spaces," vol. 567, no. 7747, pp. 209–212.

[6] J. R. Glick, T. P. Gujarati, A. D. Corcoles, Y. Kim, A. Kandala, J. M. Gambetta, and K. Temme, "Covariant quantum kernels for data with group structure," *arXiv preprint arXiv:2105.03406*, 2021.

[7] T. Fletcher, "Support vector machines explained,"

[8] M. Girolami, H. Mischak, and R. Krebs, "Analysis of complex, multidimensional datasets," vol. 3, no. 1, pp. 13–19.

[9] A. D. Corcoles, A. Kandala, A. Javadi-Abhari, D. T. McClure, A. W. Cross, K. Temme, P. D. Nation, M. Steffen, and J. M. Gambetta, "Challenges and opportunities of near-term quantum computing systems," vol. 108, no. 8, pp. 1338–1352.

[10] A. A. Bruen and M. A. Forcinito, *Cryptography, Information Theory, and Error-Correction: A Handbook for the 21st Century*. John Wiley & Sons. Google-Books-ID: fd2LtVgFzoMC.

[11] E. T. Campbell, B. M. Terhal, and C. Vuillot, "Roads towards fault-tolerant universal quantum computation," vol. 549, no. 7671, pp. 172–179. Number: 7671 Publisher: Nature Publishing Group.

[12] S. D. Sarma, "Quantum computing has a hype problem."

[13] ""quantum computing," rigetti computing. [online]. available: https://www.rigetti.com/. [accessed: 01-jul-2022]."

[14] A. Roy and D. P. DiVincenzo, "Topological quantum computing." Number: arXiv:1701.05052.

[15] N. Chetan, "Microsoft has demonstrated the underlying physics required to create a new kind of qubit."

[16] X. Mi and I. et al., "Time-crystalline eigenstate order on a quantum processor," vol. 601, no. 7894, pp. 531–536. Number: 7894 Publisher: Nature Publishing Group.

[17] F. Arute and A. et al., "Quantum supremacy using a programmable superconducting processor," vol. 574, no. 7779, pp. 505–510. Number: 7779 Publisher: Nature Publishing Group.

[18] J.-E. Park, B. Quanz, S. Wood, H. Higgins, and R. Harishankar, *Practical application improvement to Quantum SVM: theory to practice*.

[19] ChenChih-Chieh, WatabeMasaya, ShibaKodai, SogabeMasaru, SakamotoKatsuyoshi, and SogabeTomah, "On the expressibility and overfitting of quantum circuit learning," Publisher: ACM PUB27 New York, NY, USA.