



PAPER

Unlocking quantum SVM potential: optimal feature map generation and feature selection

OPEN ACCESS

RECEIVED

7 August 2024

REVISED

28 October 2024

ACCEPTED FOR PUBLICATION

12 December 2024

PUBLISHED

27 December 2024

Original content from this work may be used under the terms of the [Creative Commons Attribution 4.0 licence](#).

Any further distribution of this work must maintain attribution to the author(s) and the title of the work, journal citation and DOI.

Sumaiyah Zahid¹  and Muhammad Atif Tahir²¹ Department of Computer Science, National University of Computer & Emerging Sciences, FAST–NUCES, Karachi, Pakistan² Department of Computer Science, Institute of Business Administration, Karachi, PakistanE-mail: sumaiyah@nu.edu.pk and atifahir@iba.edu.pk**Keywords:** quantum support vector machine, kernel methods, feature selection, quantum parameterized circuits, quantum machine learning, tabu searchSupplementary material for this article is available [online](#)**Abstract**

The study proposes a mechanism to generate effective feature maps with optimal feature selection using the Tabu Search algorithm. It compares the performance of classical support vector machines (SVM), quantum support vector machines (QSVM) with only gate selection, and QSVM with both gate selection and feature selection (QSVM-FS) across various datasets. The results indicate that classical SVMs excel with several benchmark datasets, while QSVMs show superior performance on synthetic datasets with non-linear separability. Notably, QSVM-FS consistently outperforms QSVM without feature selection, highlighting the importance of feature selection in enhancing model accuracy. These findings suggest that while both quantum and classical SVMs have unique advantages, quantum methods offer particular benefits in specific scenarios. In the NISQ era, classical simulations are a primary tool for assessing quantum experiments, though they face challenges such as design impacts, limited scales, and biases. Ultimately, no definitive winner exists between quantum and classical methods, as both have their own strengths.

1. Introduction

The primary goal of quantum computing is to extend the limits of computational efficiency, speed, and accuracy. Quantum computing and its associated fields have witnessed significant advancements since the announcement of quantum supremacy [1]. This evolving field encompasses a range of tasks, including processing classical data using quantum algorithms, handling quantum data with classical algorithms, and processing quantum data with dedicated quantum algorithms. Applications of quantum machine learning involve exploring machine learning for quantum applications, developing quantum algorithms, optimizing feature maps within quantum Hilbert space, and creating quantum neural networks for classical or quantum information. The progress in this field has already produced numerous algorithms and applications such as [2, 3].

Quantum Machine Learning (QML) models use quantum circuits with adjustable parameters. Classical data is first transformed into quantum features via a feature map $|\psi(x)\rangle$ using techniques like amplitude, basis, or angle encoding. To fully exploit quantum properties such as superposition and entanglement, a parameterized unitary transformation is applied, allowing fine-tuning based on the specific problem.

Kernel methods play a crucial role in both classical and Quantum Machine Learning (QML). These methods can be employed for regression, classification, and dimensionality reduction. The achievement of a quantum advantage is feasible when quantum kernels are computationally infeasible for classical systems [4]. One of the most effective supervised kernel methods is the Support Vector Machine (SVM), which optimizes an algorithm to find the best linear hyperplane that separates data points into different classes, maximizing the minimum margin within the training dataset. To handle non-linear decision boundaries, non-linear data is mapped to a higher-dimensional feature space using a feature map, with the range of the feature space defined by this

mapping function. The Quantum Support Vector Machine (QSVM) algorithm enhances the classical SVM by utilizing a quantum circuit [5, 6], known as an ansatz, as the feature map function. The feature map is generated using a parameterized quantum circuit based on the values of the data point. This encoding enables the mapping of data points into a high-dimensional quantum state within a quantum Hilbert space. The parameters of this quantum circuit are trainable [7, 8]. Feature Selection (FS) is a critical aspect of training Machine Learning (ML) models as it significantly impacts performance. Selecting optimal features requires evaluating multiple combinations to optimize model metrics such as accuracy or efficiency. This task is particularly challenging due to the exponential time complexity and high dimensionality of datasets [9]. The recent work by Wang *et al* (2023) [10] introduces quantum-based frameworks for feature selection, which have been especially effective in domains like finance and healthcare. Tahir *et al* (2007) [11] presents a hybrid approach that applies Tabu Search for simultaneous feature selection, demonstrating its effectiveness in tasks like prostate cancer diagnosis using multispectral images. Additionally, Grossi *et al* (2006) [12] explored hybrid quantum–classical methods that improve both feature selection and classification performance. Furthermore, Liu *et al* (2023) [13] conducted a comprehensive review of quantum feature selection methods, emphasizing their ability to outperform classical techniques in high-dimensional feature spaces.

Feature Selection (FS) is another crucial aspect of training Machine Learning (ML) models because it directly influences their performance. Selecting effective features involves selecting multiple combinations to outperform a model based on a given metric. This task is challenging owing to its exponential time complexity and the complexity of highly dimensional datasets [9].

In this study, our focus is on the effective selection of feature maps to create efficient parameterized quantum circuits using the Tabu Search algorithm. Together with finding an optimized kernel, our target is to find the subset of features which help better in generating a feature map. This study aims to determine whether generating a feature map from all the features is more effective or if using a subset of them yields better results. The goal is not to outperform classical methods but to provide insights into the most beneficial feature selection for future research. Section 2 reviews Quantum Machine Learning (QML), focusing on quantum feature maps and kernels for supervised learning with support vector machines. Section 3 details the methodology for selecting optimized quantum kernels using Tabu Search, including the encoding mechanism and fitness function. Section 4 presents the results, comparing classical and quantum methods on benchmark datasets. Section 5 discusses these results, section 6 concludes the study, and section 7 outlines future directions for quantum kernel optimization.

2. Literature review

Quantum Machine Learning (QML) has gained significant attention in recent years, although its foundations were laid in the early 1990s. Early breakthroughs in quantum algorithms, such as Shor’s algorithm for prime factorization [14] and Grover’s search algorithm [15], demonstrated the potential of quantum computing to outperform classical approaches. These early advancements help QML to rapidly grow to leverage quantum computational power in order to solve complex machine learning tasks more efficiently and to achieve what is known as ‘quantum advantage.’ One of the seminal contributions in QML is the pioneering work on Quantum Support Vector Machines (QSVM) by Rebentrost *et al* (2014) [16]. This study demonstrated how quantum circuits can be used for big data classification, which also helps in advancements of many subsequent applications of quantum-enhanced machine learning algorithms. QSVM utilizes quantum kernels to handle large datasets more efficiently than classical SVMs, showing an important leap in the exploration of quantum algorithms for machine learning. Recent research in QML has explored a range of quantum-enhanced machine learning models, algorithms, and their applications. A comprehensive review by Zeguedry *et al* (2023) [17] compares quantum machine learning algorithms with their classical counterparts to demonstrate the potential of quantum-enhanced algorithms in fields such as handwritten digit recognition and medical diagnosis. The study implemented Quantum Neural Networks (QNNs) and Quantum Support Vector Machines (QSVMs) to demonstrate their advantages in addressing specific domain problems like breast cancer diagnosis. The role of quantum contextuality in machine learning was explored by Bowles *et al* (2023) [18]. The research emphasizes how quantum non-classical features, such as contextuality, can enhance the inductive bias of machine learning models. This study is significant as it explores the direct application of quantum principles to improve model generalization and expressivity. QML’s potential has been demonstrated across various domains, including image classification and medical diagnostics. Senokosov *et al* (2023) [19] proposed hybrid quantum–classical neural networks with quantum layers, significantly improving accuracy on benchmark datasets like MNIST and CIFAR-10 while reducing the number of parameters compared to classical models. Similarly, Wei *et al* (2023) [20] reviewed several quantum methods for medical image analysis, emphasizing how quantum algorithms could improve diagnostic accuracy and computational efficiency. Melnikov *et al* (2023)

[21] contributed by focusing on the intersection of quantum technology and artificial intelligence for improving classical machine learning solutions. Feature selection remains a critical aspect of machine learning that directly influence model performance. In QML, feature selection involves identifying optimal quantum feature maps that effectively transform classical data into quantum states. Mücke *et al* (2022) [9] explored feature selection for quantum models, noting that selecting the right subset of features can drastically enhance the performance of QSVM. This approach is particularly relevant for high-dimensional data, where feature selection can mitigate the exponential time complexity challenges faced by quantum algorithms. QML has also made significant contribution in fields such as disease diagnosis [22] which highlighted the use of quantum algorithms in diagnosing neurodegenerative diseases. Moreover, Khurana *et al* (2023) [23] reviewed state-of-the-art advancements in quantum-enhanced natural language processing, focusing on key challenges and trends in natural language processing, while Horak and Sablatnig (2019) [24] provided an overview of quantum deep learning models for image recognition, showcasing their efficacy in various visual tasks.

2.1. Quantum kernel methods

A parameterized quantum circuit, also known as a variational circuit or quantum neural network, contains tunable parameters. As we increase the depth and number of gates in the quantum circuit, the accuracy also tends to improve. However, it is crucial to develop circuits that can optimize our objective using a minimum number of parameters, as hardware noises and errors also increase with complexity. Since finding the optimal circuit is quite difficult, Herrman *et al* (2022) [25] showed that a good set of parameters can be found in polynomial time. Cerezo *et al* (2022) [26] claims that variational approaches have the potential to provide a quantum advantage by finding the optimal circuit that can filter signals from noise, leading to improved accuracies. The flexibility in the structure of quantum circuits allows them to represent many quantum states efficiently.

Quantum circuits are also considered as kernel methods in machine learning where the data is process in high-dimensional Hilbert spaces to carryout powerful data analysis and learning capabilities. Schuld (2021) [5] provides a comprehensive overview of how quantum machine models can be formalized as kernel methods. The study elaborates that quantum models involves data in high dimensional spaces, and this approach guarantees that kernel-based training helps in finding better or with same equivalence models in comparison of variational circuit training. The benefit of using tensor networks (TNs) mapped to parameterized quantum circuits (PQCs) in order to solve complex quantum simulation problems by decomposing matrix product states (MPS) into shallow quantum circuits is explored by Rudolph *et al* (2023) [27]. Through this approach, the study demonstrates the performance of TN algorithms by approximating TN states using realistic quantum circuits. Ragone (2023) [28] helps in understanding and mitigating sources of barren plateaus in quantum circuits by providing the exact expression for the variance of the loss function using deep parameterized quantum circuits. In addition, Cerezo *et al* (2021) [29] contributes by investigating a significant importance of selecting cost function that impacts the trainability of quantum circuits. Thanasilp *et al* (2024) [30] provides the performance of quantum kernel models by focusing on obtained resources for precise kernel value estimation. Their findings suggests to avoid features leading to exponential concentration, ensuring efficient evaluation and performance of quantum kernel methods. Moreover, Foss *et al* (2023) [31] demonstrates the advantage of adaptive quantum circuits in comparison to purely unitary quantum circuits in order to achieve tasks employ mid-circuit measurements and classical computations. The results highlights the superior performance of adaptive quantum circuits in constructing long-range entangled topological states. Jager *et al* (2023) [32] demonstrated the efficiency of variational quantum circuits as solvers for BQP (Bounded Error Quantum Polynomial Time) problems.

2.2. Quantum feature selection

There are several different approaches proposed for advancement in quantum feature selection and its potential impact. Grossi *et al* (2022) [33] emphasized the importance of feature selection through Quantum Support Vector Machine (QSVM) with a hybrid quantum–classical approach for fraud detection. While Albino *et al* (2023) [34] introduced a heuristic approach called as Evolutionary Quantum Feature Selection (EQFS) which employs the Quantum Circuit Evolution (QCE). The EQFS demonstrates quadratic scaling in the number of features which offers a promising solution for efficient feature selection in complex datasets. Wang *et al* (2023) [35] propose a quantum feature selection framework using Hamiltonian encoding and a ground state preparation algorithm to achieve the same or better classification accuracy compared to the original data without feature selection. Poggiali *et al* (2023) [36] also highlights the potential of quantum algorithms to enhance feature selection by proposing the Hybrid Quantum Feature Selection (HQFS) algorithm to perform of synthetic and real datasets. Mücke *et al* (2022) [37] propose another feature selection algorithm based on Quadratic Unconstrained Binary Optimization (QUBO), which allows the selection of specified number of

features based on their importance and redundancy. The QUBO approach provides the higher-quality solutions compared to iterative and greedy methods. These studies provide the unique capabilities of quantum computing to address the challenges of feature selection with continuous improvements in the accuracy and efficiency of machine learning models.

2.3. Quantum SVM and Tabu Search

Quantum Support Vector Machines (QSVM) leverage the principles of quantum computing to enhance the performance and accuracy in comparison to traditional Support Vector Machines (SVM). Park *et al* (2020) [38] proposed the use of quantum feature maps to translate data into quantum states to build the SVM kernel out of these states. Their study elaborates competitive performance of QSVM across several datasets, outperforming classical SVMs with radial basis function (RBF) kernels. Shaik *et al* (2022) [39] explored the accuracy and processing speed trade-offs in QSVM classifiers using PRISMA hyperspectral imagery that shows QSVM offers higher accuracy than traditional SVM. While Houssein *et al* (2021) [40] introduced a hybrid quantum-kernel support vector machines (QKSVM) approach combined with Binary Harris hawk optimization (BHHO) for gene selection and QSVM for cancer classification which demonstrates enhanced performance.

Evolutionary algorithms have the ability to work on multi-objective problems effectively since they are able to generate optimized solutions. To optimize quantum circuits for a feature map generation evolutionary algorithms are very useful [41]. A detailed implementation of NSGA-II for automating the design of feature map circuits has been presented using QSVM classifiers for tabular data [42]. The primary goal was to achieve a balance between maximizing classification accuracy and minimizing circuit size. Study claims a perfect accuracy on the dataset and shows supremacy over classical algorithms. Tabu Search (TS) is a meta-heuristic optimization technique that enhances local search methods by using memory structures to avoid cycling and improve efficiency. It is particularly effective for combinatorial and NP-hard problems. Alotaibi *et al* (2022) [43] proposed a TS-based data clustering algorithm with adaptive Search Memory (ASM) that combines TS with K-means clustering, demonstrating superior performance over other techniques. Zhang *et al* [44] integrate TS with a flow allocation model for real-time scheduling of autonomous mining trucks, optimizing mining operations. Suganthi *et al* [45] introduce a multi-swarm optimization (MSO) approach combined with TS to enhance network lifetime and energy efficiency in wireless sensor networks, outperforming other methods.

The integration of QSVM and TS can potentially offer a robust framework to address complex machine learning and optimization problems. QSVM provides the computational efficiency and enhanced classification capabilities of quantum computing, while TS offers an effective optimization technique for gate selection and feature selection and model tuning. Our study is inspired by the method presented by Altaraes *et al* (2021) [42]. However, we observed that the authors reported perfect accuracy by dividing the dataset into only training and test sets. To ensure better generalization, we maintained independent training, validation, and test datasets in our approach. Additionally, our Tabu Search fitness function operates solely on the validation dataset, ensuring that the results reported here are unbiased by this process.

3. Methodology

In the NISQ (Noisy Intermediate-Scale Quantum) era, one of the major challenges is efficiently encoding classical data into quantum states to bring out quantum advantages. Adding more depth and gates to your quantum-parameterized circuits comes with increased accuracy. Finding better accuracy with minimum gates and depth is a multi-objective optimization problem. Iterative heuristics such as genetic algorithms, Tabu search etc are quite famous techniques to solve problems related to the curse of dimensionality in traditional classification and regression methods.

3.1. Encoding

Our proposed encoding scheme for Tabu Search (TS) design comprises two primary components:

3.1.1. Gate search

The encoding method is about representing different quantum gates like Hadamard, CNOT, and Rotation gates, along with their specific angles. Each kind of gate is represented by a single letter. This way of encoding creates a string of letters, and its length is found by multiplying how deep the system is with the number of qubits. The figure 1 shows how letters are connected to their corresponding gates. The length of the string is just the depth multiplied by the number of qubits.

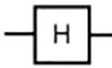
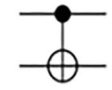
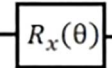
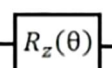
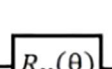
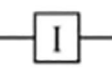
| Character Encoding | Gate | Angle | Symbol |
|--------------------|---------------------------|---------|---|
| A | Hadamard | NA |  |
| B | Controlled Not (CNOT, CX) | NA |  |
| C | Rotation X Rx | π |  |
| D | | $\pi/2$ | |
| E | | $\pi/4$ | |
| F | | $\pi/8$ | |
| G | Rotation Z Rz | π |  |
| H | | $\pi/2$ | |
| I | | $\pi/4$ | |
| J | | $\pi/8$ | |
| K | Rotation Y Ry | π |  |
| L | | $\pi/2$ | |
| M | | $\pi/4$ | |
| N | | $\pi/8$ | |
| O | Identity | NA |  |
| P | | | |
| Q | | | |
| R | | | |

Figure 1. Encoding scheme for gates.

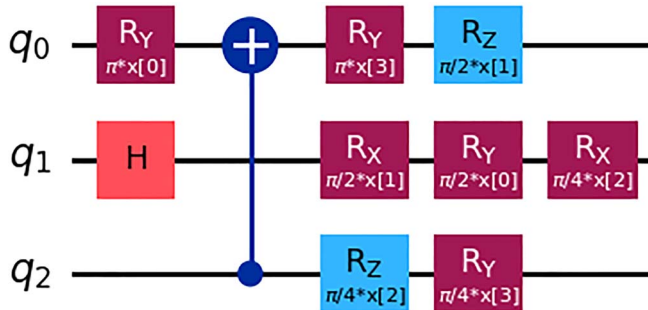


Figure 2. Quantum circuit based on encoded string.

3.1.2. Feature search

In our encoding approach, the secondary aspect highlights a binary representation. Here, the existence of a particular feature is indicated by '1', while its non-existence is indicated by '0'. For instance, if we have a dataset with 6 features and the encoded feature string is '110101', it defined as features 1, 2, 4, and 6 are considered for analysis, while features 3 and 5 are currently excluded from consideration.

3.2. Initial solution

The figure 2 represents the features of the dataset as $x[0], x[1], \dots, x[n]$, where 'n' is the total number of features. These features are utilized to generate the quantum feature map. The values $x[0], x[1]$, and so on, referring to the classical data that serves as parameters for encoding in the quantum ansatz. The initial stage, specifically directed towards the Gate Search component, commences with a randomized initialization procedure. Subsequently, in the second phase concerning Feature Encoding, all extant features are incorporated by assigning a binary value of '1' to each feature. This establishes an initial state encompassing the entirety of potential features, laying the groundwork for subsequent analytical processes and refinement. The string is generated from a random combination of gates, defined by the hyperparameters for the number of qubits and circuit depth, resulting in a 12-character string ($4 \text{ qubits} \times 3 \text{ depth} = 12 \text{ gates}$). This string consists of two parts: the Gate Search and Feature

Encoding. The specific string ‘KABRDIKLPHEM1111’ is produced during the Gate Search phase through randomized initialization. The final segment, ‘1111,’ indicates that all features are included in the initial solution, with each ‘1’ signifying the presence of a feature. For example, for a dataset with four features, a number of qubits of 3 and a circuit depth of 4, let’s have an initial string ‘KABRDIKLPHEM1111’. It will create an initial circuit like in figure 2.

3.3. Neighborhood solutions

To generate a neighboring solution, we randomly change one character and either add or delete one feature. For instance, starting with the string ‘KABRDIKLPHEM1111’ two neighboring solutions could be ‘KACRDIKLPHEM1101’ and ‘KABRDIKAPHEM1110.’ In the first solution, the character ‘B’ is replaced with ‘C,’ and the third feature is deleted. In the second solution, the ‘L’ at index 6 is replaced with ‘A,’ and the fourth feature is deleted.

3.4. Tabu moves

Our process involves maintaining two Tabu lists. The first list corresponds to characters representing gate selection. If a gate is changed at any index, then changing that specific gate with another is considered Tabu for the next n iterations, and the index is recorded in the Tabu list. The second Tabu list is dedicated to feature addition or deletion. If a feature has been either added or removed, subsequent iterations attempting to add or remove that particular feature are considered Tabu.

Algorithm 1. QSVM—Tabu Search Algorithm

```

1: Initialize the gate string  $a_1$ 
2: Initialize the feature string  $a_2$ 
3: Merge gate and feature  $a_1$  and  $a_2$  into  $s_0$ 
4: Initialize the current solution  $s$  with an initial solution  $s_0$ 
5: Set the best solution  $s_{best} \leftarrow s$ 
6: Initialize the tabu list  $TL$  to an empty list
7: Set the maximum number of iterations  $max\_iterations$ 
8: for  $i = 1$  to  $max\_iterations$  do
9:   Generate a set of candidate solutions  $N(s)$  from the neighborhood of  $s$  without taking a tabu move
10:  Add solutions in  $N(s)$  that satisfy the Aspiration criteria
11:  Generate quantum feature map
12:  Train the QSVM
13:  Evaluate accuracy on validation data
14:  Select the best candidate solution  $s' \in N(s)$ 
15:  if  $s'$  is better than  $s_{best}$  then
16:     $s_{best} \leftarrow s'$ 
17:  end if
18:   $s \leftarrow s'$ 
19:  Add the move to the tabu list  $TL$ 
20:  Remove expired moves from  $TL$ 
21: end for
22: return  $s_{best}$ 
23: Evaluate accuracy on test data

```

3.5. Fitness function

In our approach, we define a fitness function to select the best candidate in the Tabu Search algorithm. The fitness function is designed to evaluate and compare candidates based on the number of features selected (cost) and their classification accuracy. The selection criteria are as follows:

1. A candidate is considered better if it selects fewer features and its validation accuracy is at least as good as the current best candidate.
2. If two candidates select the same number of features, the one with higher validation accuracy is preferred.

$$\text{Accuracy} = \frac{\text{Number of correctly classified instances}}{\text{Total number of instances}} \quad (1)$$

This ensures that the candidate either selects fewer features without sacrificing accuracy or achieves better accuracy with the same number of features.

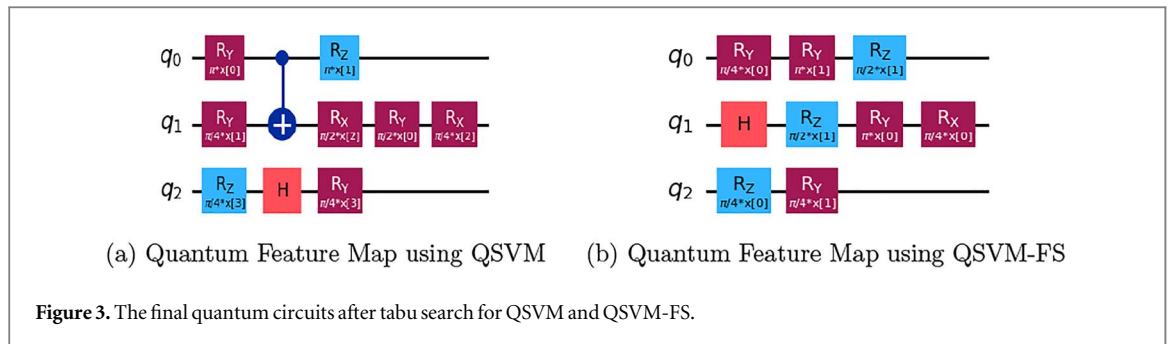


Figure 3. The final quantum circuits after tabu search for QSV and QSV-FS.

3.6. Aspiration criterion

The aspiration criterion serves as a mechanism to temporarily bypass the Tabu status of moves when it is suitable. It allows for the temporary override of Tabu restrictions if a move demonstrates significant merit. In our approach, when a feature is added, deleted, or a gate has been changed at a particular index during iteration i , and this action leads to the best cost across all preceding iterations, the feature is permitted to be added, deleted, or gate to be modified, even if it is listed in the Tabu list 3(a).

3.7. Quantum feature map

For QSV, the string KABRDIKLPHEM is specified, and all features are automatically selected. After completing the Tabu search for QSV, the final solution obtained is KMQBDIPLAGEM. The resulting quantum ansatz is illustrated in figure For QSV-FS, the string KABRDIKLPHEM1111 is specified, and all features are selected initially. After completing the Tabu search for QSV-FS, we obtain the final solution, MARRHIKPPHEM0011. To generate a feature map from this solution, we will use only the second and third features of the dataset (with their column numbers reset to 0 and 1, respectively). This provides the Quantum ansatz as shown in figure 3(b)

3.8. Quantum SVM

Support Vector Machine (SVM) is a widely used classification algorithm designed to find a hyperplane that can effectively separate data points in a high-dimensional space. In our implementation, we employ two types of kernels: the Linear kernel and the Radial Basis Function (RBF) kernel. These choices are inspired by the classical SVM kernels, known for their effectiveness in various classification tasks.

The Linear kernel provides a straightforward approach to classification by assuming linear separability of the data, while the RBF kernel, known for its superior performance, captures non-linear relationships by mapping the data into a higher-dimensional space. By incorporating both the Linear and RBF kernels, we aim to evaluate and replicate the performance of these classical kernels within the framework of Quantum SVM.

The Quantum Linear Kernel function is defined as shown in equation (2). Where, $K_{\text{Linear}}(\psi(x), \psi(x'))$ is the Quantum Linear Kernel function. $\psi(x)$ and $\psi(x')$ are the quantum feature maps of the input feature vectors x and x' . $\langle \psi(x), \psi(x') \rangle$ denotes the inner product of the quantum feature maps $\psi(x)$ and $\psi(x')$.

$$K_{\text{Linear}}(\psi(x), \psi(x')) = |\langle \psi(x) | \psi(x') \rangle| \tag{2}$$

The Radial Basis Function (RBF) kernel is defined as in equation (3). Where, $K_{\text{RBF}}(\psi(x), \psi(x'))$ is the Quantum RBF kernel function. $\psi(x)$ and $\psi(x')$ are the quantum feature maps of the input feature vectors x and x' . γ is a parameter that determines the spread of the kernel. $\|\psi(x) - \psi(x')\|$ represents the Euclidean distance between the quantum feature maps $\psi(x)$ and $\psi(x')$.

$$K_{\text{RBF}}(\psi(x), \psi(x')) = \exp(-\gamma \|\psi(x) - \psi(x')\|^2) \tag{3}$$

The study proposed Tabu search which does two searches simultaneously i.e. gate selection and feature selection to find efficient feature map with optimal feature sets as defined in algorithm 1 and shown in figure 4. For this we divided our encoding string into two parts respectively.

4. Experiments

The datasets involved in the experiments have been divided into training, validation, and testing ratio of 80% 10%, and 10%. To mitigate the risk of overfitting, Tabu optimization is exclusively carried out on the training data, with fitness computed on a separate validation dataset. This separation ensures that the model is not bias or excessively tailored to the training data. The completion of all iterations and optimization processes, the final evaluation is then followed by testing on an independent test dataset. This approach helps to ensure a more

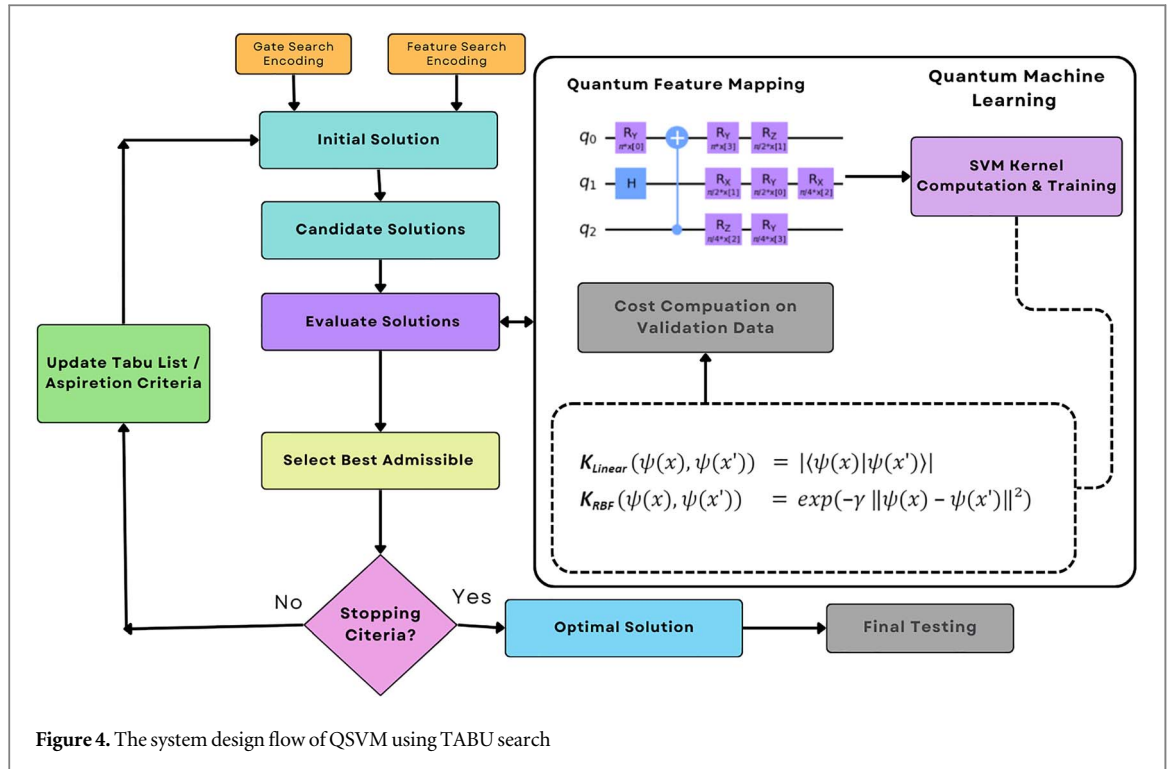


Table 1. The details of datasets involved in the experiments.

| Dataset | Samples | Features | Classes |
|--------------------|---------|----------|---------|
| Ionosphere | 351 | 34 | 2 |
| Parkinson | 195 | 22 | 2 |
| Statlog Australian | 690 | 14 | 2 |
| Statlog Heart | 270 | 13 | 2 |
| Wine | 178 | 13 | 3 |
| Iris | 150 | 4 | 3 |
| Moon | 500 | 2 | 2 |
| Circles | 500 | 2 | 2 |

robust and generalized performance. The experiments carried by executing fifty independent runs to generalize the results.

4.1. Datasets

Table 1 describes the details of the datasets which have been used for the experiments followed by the comparison of algorithms. The datasets are not transformed in the carried-out experiments, such as using PCA can eliminate features and affect the authenticity of the data, leading to the overfitting of the models as highlighted in [46].

4.2. Experiments

The study carried out several experiments to provide the performance comparison of the linear and RBF kernels in classical and Quantum approach.

4.2.1. Comparison of linear kernel of classical SVM and QSVM

In this experiment, QSVM linear kernel was calculated initially using the gate selection only followed by using gate selection and feature selection which reduced the features. The achieved results were compared with classical SVM with linear kernel based on obtained accuracy. Table 2 shows the calculations achieved in this experiment where the classical SVM linear kernel is defined as ‘SVM’, the QSVM linear kernel with gate selection only defined as ‘QSVM’ and the QSVM linear kernel with gate selection as well as feature selection is defined as ‘QSVM—FS’ (FS representing Feature Selection) respectively. The values marked as ‘NA’ in table 2 indicate

Table 2. Performance comparison between the Quantum Linear Kernel used for QSVM, QSVM with Feature Selection, and Classical SVM with a Linear Kernel. Accuracy is represented as mean \pm standard deviation. QSVM-FS shows the accuracy and the mean of feature counts.

| Dataset | QSVM | QSVM-FS | | | | Classical SVM |
|------------|--------------------|----------------------------|---------------------------|----------------------------|----------------------------|--------------------|
| | | Tabu Search | Random Search | Bayesian Optimization | Genetic Search | |
| Ionosphere | 0.8799 \pm 0.065 | 0.8821 \pm 0.055 (16.96) | 0.8666 \pm 0.053 (12.9) | 0.8717 \pm 0.061 (16.06) | 0.8655 \pm 0.052 (15.48) | 0.8781 \pm 0.059 |
| Parkinsons | 0.8244 \pm 0.063 | 0.8298 \pm 0.073 (5.72) | 0.8015 \pm 0.073 (6.52) | 0.8127 \pm 0.078 (6.78) | 0.8196 \pm 0.071 (8.3) | 0.8529 \pm 0.082 |
| Australian | 0.7475 \pm 0.109 | 0.8467 \pm 0.051 (6.8) | 0.8400 \pm 0.059 (4.06) | 0.8417 \pm 0.047 (5.8) | 0.8428 \pm 0.045 (5.78) | 0.8536 \pm 0.047 |
| Heart | 0.6896 \pm 0.108 | 0.7430 \pm 0.089 (5.6) | 0.739 \pm 0.105 (3.8) | 0.737 \pm 0.104 (5.38) | 0.7385 \pm 0.095 (5.48) | 0.8281 \pm 0.060 |
| Wine | 0.863 \pm 0.087 | 0.871 \pm 0.084 (3.8) | 0.8912 \pm 0.086 (4.08) | 0.9215 \pm 0.065 (5.2) | 0.8948 \pm 0.097 (5.42) | 0.9595 \pm 0.044 |
| Iris | 0.9241 \pm 0.061 | 0.9600 \pm 0.053 (2.5) | 0.948 \pm 0.066 (1) | 0.9486 \pm 0.061 (1.5) | 0.9453 \pm 0.068 (1.46) | 0.9267 \pm 0.034 |
| Moons | 0.9708 \pm 0.031 | NA | NA | NA | NA | 0.8616 \pm 0.049 |
| Circles | 0.8544 \pm 0.058 | NA | NA | NA | NA | 0.5228 \pm 0.048 |

”Not Applicable,” where feature selection was not performed due to the datasets already having a minimal number of features.

4.2.2. *Comparison of RBF kernel of classical SVM and QSVM*

This experiment calculates the RBF Kernel of QSVM by using the gate selection initially followed by feature selection and gate selection which reduced the features involved in this experiment. The results obtained were compared based with classical SVM with RBF kernel on obtained accuracy. Table 3 shows the obtained calculations where ‘QSVM’ represents the calculations of QSVM RBF Kernel. The ‘QSVM—FS’ represents the obtained accuracy with reduced feature selection to calculate QSVM RBF Kernel and ‘SVM’ defines the classical SVM RBF kernel calculations respectively. The values marked as ‘NA’ in table 3 indicate ‘Not Applicable,’ where feature selection was not performed due to the datasets already having a minimal number of features.

5. Results

The results of Experiment 1 are presented in table 2, which shows the performance comparison between the linear kernels of classical SVM, Quantum-SVM, and Quantum-SVM with feature selection (or reduced features). The performance comparison in table 2 indicates that the classical SVM linear kernel outperforms the other models for almost half of the datasets, such as Parkinsons, Australian, Heart, and Wine, demonstrating an advantage over QSVM and QSVM-FS. On the other hand, QSVM and QSVM-FS showed better performance on Ionosphere, Iris, Moons, and Circles. Further analysis reveals that among the different feature selection methods used for QSVM-FS, Tabu Search consistently provided the best balance between accuracy and feature reduction across datasets like Ionosphere, Parkinsons, Australian, and Iris. For example, in the Ionosphere dataset, Tabu Search achieved the highest accuracy (0.8821) with 16.96 selected features, outperforming Bayesian Optimization and Random Search. Meanwhile, Bayesian Optimization exhibited the best performance for the Wine dataset (0.9215), indicating its suitability for certain types of datasets. On the contrary, Random Search, despite reducing feature counts significantly, generally resulted in lower accuracy, as seen in datasets like Australian and Iris. Thus, feature selection approaches greatly influence the performance of QSVM-FS, with Tabu Search and Bayesian Optimization offering the best results in most cases. Furthermore, QSVM-FS consistently shows improved performance over basic QSVM across nearly all datasets considered in this study. Overall, the performance comparison indicates that there isn’t a clear winner between quantum and classical SVM as they serve different purposes, with each having strengths in specific datasets. Additionally, QSVM with Feature Selection suggests that feature selection can enhance the model’s accuracy.

The RBF kernel results from Experiment 2 are presented in table 3, comparing the performance between classical SVM, QSVM, and QSVM with Feature Selection. The results indicate that the classical SVM RBF kernel outperforms QSVM and QSVM-FS for most of the datasets, including Ionosphere, Parkinsons, Australian, Heart, Wine, and Iris. However, QSVM-FS demonstrates robust performance compared to basic QSVM, highlighting the importance of feature selection in improving performance and accuracy. QSVM outperforms the classical SVM RBF kernel on synthetic datasets like Moons and Circles.

The RBF kernel results from Experiment 2 are presented in table 3, comparing the performance between classical SVM, QSVM, and Feature selection techniques like Tabu Search, Random Search, Bayesian Optimization, and Genetic Search play an essential role by optimizing the feature subset size and enhancing QSVM’s results. Tabu Search often yields the best performance, as seen in the Australian dataset, achieving 0.8318 accuracy, compared to 0.7581 for plain QSVM. Bayesian and Genetic Search also achieve competitive results, although Random Search is prone to performance drops (e.g., Ionosphere: 0.6427 accuracy). These techniques also control feature selection, with fewer features improving efficiency but sometimes sacrificing accuracy. While feature selection enhances QSVM performance, classical SVM still leads in most cases. However, on synthetic datasets like Moons and Circles, QSVM performs slightly better or comparably. For example, QSVM achieves 0.9881 on Moons, slightly outperforming classical SVM (0.9868). Feature selection was not applied to Moons and Circles (indicated by NA), because these datasets have fewer features. The comparative analysis reveals that Tabu Search (QSVM-FS) demonstrates strong performance in most cases, striking a balance between accuracy and feature reduction. In several datasets, Tabu Search not only matches but often surpasses the performance of other quantum search methods, such as Random Search, Bayesian Optimization, and Genetic Search, indicating its capability to efficiently explore complex feature spaces and avoid local optima. Tabu Search exhibits consistent results across datasets, achieving accuracy levels comparable to or better than Random Search and Bayesian Optimization while selecting a similar or slightly larger number of features. This shows that while Bayesian Optimization might select fewer features in certain cases, it does not always translate into improved accuracy. In datasets like Ionosphere and Parkinsons, Tabu Search delivers

Table 3. Performance comparison between the Quantum linear kernel used for QSVM, QSVM with Feature Selection, and classical SVM with a linear kernel. Accuracy is represented as meanstandard deviation. QSVM-FS shows the accuracy and the mean of feature counts.

| Dataset | QSVM | QSVM-FS | | | | Classical SVM |
|------------|--------------------|--------------------------|---------------------------|---------------------------|---------------------------|--------------------|
| | | Tabu Search | Random Search | Bayesian Optimization | Genetic Search | |
| Ionosphere | 0.7899 ± 0.044 | 0.796 ± 0.084 (19.6) | 0.6427 ± 0.017 (9.86) | 0.6465 ± 0.055 (11.5) | 0.6678 ± 0.053 (11.8) | 0.9413 ± 0.035 |
| Parkinsons | 0.7654 ± 0.025 | 0.7781 ± 0.017 (2.9) | 0.7539 ± 0.018 (5.28) | 0.755 ± 0.019 (5.46) | 0.754 ± 0.018 (6.7) | 0.8775 ± 0.064 |
| Australian | 0.7581 ± 0.072 | 0.8318 ± 0.047 (5.0) | 0.8292 ± 0.065 (3.94) | 0.8248 ± 0.049 (5.58) | 0.8318 ± 0.067 (6.8) | 0.8548 ± 0.048 |
| Heart | 0.662 ± 0.094 | 0.7372 ± 0.095 (4.9) | 0.7259 ± 0.077 (3.62) | 0.7155 ± 0.095 (5.02) | 0.7088 ± 0.095 (5.75) | 0.8393 ± 0.070 |
| Wine | 0.8233 ± 0.126 | 0.848 ± 0.098 (5.3) | 0.8607 ± 0.099 (4.34) | 0.865 ± 0.085 (5.76) | 0.8572 ± 0.108 (5.73) | 0.9801 ± 0.031 |
| Iris | 0.9431 ± 0.085 | 0.9524 ± 0.068 (2.5) | 0.949 ± 0.058 (1.1) | 0.9426 ± 0.069 (1.5) | 0.9413 ± 0.069 (1.4) | 0.9600 ± 0.049 |
| Moons | 0.9881 ± 0.022 | NA | NA | NA | NA | 0.9868 ± 0.015 |
| Circles | 0.8802 ± 0.049 | NA | NA | NA | NA | 0.8704 ± 0.044 |

competitive accuracy, demonstrating that the added computational sophistication of the algorithm can lead to meaningful improvements over more stochastic methods.

The results demonstrate that that QSVM excels on classification tasks where the data is not easily separable. Additionally, the study highlights that QSVM with feature selection consistently provides better performance than basic QSVM and achieves performance comparable to classical SVM kernels on several datasets.

6. Conclusion

Quantum Machine Learning (QML) aims to address challenges that classical methods struggle with and has the potential to significantly impact the future of AI. This study focused on developing techniques for automatically creating quantum ansatz from classical data. The goal was to generate quantum feature maps that enhance the performance of quantum algorithms while maintaining simplicity and accuracy.

Our results show that classical data can pose limitations for some quantum ML applications. QML excels with data that is highly non-linearly separable, particularly in classification tasks. Comparing quantum and classical methods directly is complex, as QML is designed to tackle problems that classical methods may not address effectively [46]. In particular, evaluating the performance of QML on simpler datasets, such as those with low dimensionality or few features, may not fully demonstrate its advantages, especially when methods like feature selection are involved. Thus, evaluating these methods solely on accuracy with classical data may not provide a fair or comprehensive comparison.

In summary, developing universal techniques for the automatic optimization of quantum feature maps across various datasets is a substantial challenge. While the QSVM-FS method shows promise in specific cases, its full potential may be realized on more complex datasets with higher dimensionality, where classical methods may struggle. By identifying dataset types, patterns, and characteristics across different problems, and by formulating strategies for parameter fine-tuning, we can gain crucial insights that will advance the field of quantum machine learning and enhance algorithm efficiency.

Data availability statement

The data that support the findings of this study are openly available at the following URL/DOI: <https://doi.org/Ionosphere>: <https://archive.ics.uci.edu/dataset/52/ionosphere> Parkinsons: <https://archive.ics.uci.edu/dataset/174/parkinsons> Statlog Australian: <https://archive.ics.uci.edu/dataset/143/statlog+australian> +credit+approval Statlog Heart: <https://archive.ics.uci.edu/dataset/145/statlog+heart> Wine: <https://archive.ics.uci.edu/dataset/109/wine> Iris : <https://archive.ics.uci.edu/dataset/53/iris> Moons : https://scikit-learn.org/stable/modules/generated/sklearn.datasets.make_moons.html Circles : https://scikit-learn.org/1.5/modules/generated/sklearn.datasets.make_circles.html.

ORCID iDs

Sumaiyah Zahid  <https://orcid.org/0000-0002-7138-8482>

References

- [1] Arute F, Arya K, Babbush R, Bacon D, Bardin J C, Barends R, Biswas R, Boixo S, Brandao F G and Buell D A E A 2019 *Nature* **574** 505–10
- [2] Biamonte J, Wittek P, Pancotti N, Rebentrost P, Wiebe N and Lloyd S 2017 *Nature* **549** 195–202
- [3] Schuld M and Petruccione F 2018 *Supervised Learning with Quantum Computers* (Springer) p 17
- [4] Wu Y, Wu B, Wang J and Yuan X 2023 *Quantum* **7** 981
- [5] Schuld M 2021 arXiv:2101.11020
- [6] Park S, Park D K and Rhee J K K 2023 *Sci. Rep.* **13** 3288
- [7] Hubregtsen T, Wierichs D, Gil-Fuster E, Derks P J H S, Faehrmann P K and Meyer J J 2022 *Phys. Rev. A* **106** 042431
- [8] Du Y, Huang T, You S, Hsieh M H and Tao D 2022 *Npj Quantum Information* **8** 62
- [9] Mücke S, Heese R, Müller S, Wolter M and Piatkowski N 2023 *Quantum Machine Intelligence* **5** 11
- [10] Wang L, Chen Z Y, Le F Y and Yu Z Q E A 2023 *Phys. Scr.* **98** 065201
- [11] Tahir M A, Bouridane A and Kurugollu F 2007 *Pattern Recognit. Lett.* **28** 438–46
- [12] Grossi M, Ibrahim N and Radescu V 2006 *Pattern Recognit. Lett.* **27** 1824–31
- [13] Liu J, Zhang W and Hu X 2023 *Journal of Quantum Computing* **9** 123–45
- [14] Shor P W arXiv:quant-ph/9508027
- [15] Grover L K 1996 A fast quantum mechanical algorithm for database search *Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing* pp 212–9
- [16] Rebentrost P, Mohseni M and Lloyd S 2014 *Phys. Rev. Lett.* **113** 130503
- [17] Zeguendry A, Jarir Z and Quafafou M 2023 *Entropy* **25** 287
- [18] Bowles J, Wright V J, Farkas M, Killoran N and Schuld M 2023 arXiv:2302.01365

- [19] Senokosov A, Sedykh A, Sagingalieva A, Kyriacou B and Melnikov A 2023 *Machine Learning: Science and Technology* **5** 015040
- [20] Wei L, Liu H, Xu J, Shi L, Shan Z, Zhao B and Gao Y 2023 *Neurocomputing* **525** 42–53
- [21] Melnikov A, Kordzanganeh M, Alodjants A and Lee R K 2023 *Advances in Physics: X* **8** 2165452
- [22] Myszczyńska M A, Ojames P N, Lacoste A M, Neil D, Saffari A, Mead R, Hautbergue G M, Holbrook J D and Ferraiuolo L 2020 *Nature Reviews Neurology* **16** 440–56
- [23] Khurana D, Koli A, Khatter K and Singh S 2023 *Multimedia Tools Appl.* **82** 3713–44
- [24] Horak K and Sablatnig R 2019 Deep learning concepts and datasets for image recognition: overview *Eleventh international conference on digital image processing (ICDIP 2019) (SPIE)* vol 11 179 pp 484–91
- [25] Herrman R, Lotshaw P C, Ostrowski J, Humble T S and Siopsis G 2022 *Sci. Rep.* **12** 6781
- [26] Cerezo M, Verdon G, Huang H Y, Cincio L and Coles P J 2022 *Nature Computational Science* **2** 567–76
- [27] Rudolph M S, Chen J, Miller J, Acharya A and Perdomo-Ortiz A 2023 *Quantum Science and Technology* **9** 015012
- [28] Ragone M, Bakalov B N, Sauvage F, Kemper A F, Marrero C O, Larocca M and Cerezo M 2023 arXiv:2309.09342
- [29] Cerezo M, Sone A, Volkoff T, Cincio L and Coles P J 2021 *Nat. Commun.* **12** 1791
- [30] Thanasilp S, Wang S, Cerezo M and Holmes Z 2024 *Nat. Commun.* **15** 5200
- [31] Foss-Feig M, Tikku A, Lu T C, Mayer K and Iqbal M E A 2023 arXiv:2302.03029
- [32] Jäger J and Krems R V 2023 *Nat. Commun.* **14** 576
- [33] Grossi M, Ibrahim N, Radescu V, Loredo R, Voigt K, Von Altrock C and Rudnik A 2022 *IEEE Transactions on Quantum Engineering* **3** 1–12
- [34] Albino A S, Pires O M, Noobath M Q and Nascimento E G 2023 arXiv:2303.07131
- [35] Wang L, Chen Z Y, Le F Y, Yu Z Q, Xue C, Zhuang X N, Yan Q, Yang Y, Wu Y C and Guo G P 2023 *Phys. Scr.* **98** 115121
- [36] Poggiali A *et al* 2023 Quantum feature selection with variance estimation *Proc of ESANN (Bruges, Belgium, 4-6 October 2023)* pp 245–250
- [37] Mücke S, Heese R, Müller S, Wolter M and Piatkowski N 2022 *Quantum Machine Intelligence* **4** 125–35
- [38] Park J E, Quanz B, Wood S, Higgins H and Harishankar R 2020 arXiv:2012.07725
- [39] Shaik R U and Periasamy S 2022 *Int. J. Remote Sens.* **43** 6176–94
- [40] Houssein E H, Abohashima Z, Elhoseny M and Mohamed W M 2021 An efficient binary harris hawks optimization based on quantum svm for cancer classification tasks *The II International Conference on Distributed Sensing and Intelligent Systems (ICDSIS 2021) (IET)* vol 2021 pp 247–58
- [41] Hakemi S, Houshmand M, KheirKhah E and Hosseini S A 2024 *Evolutionary Intelligence* **17** 627–42
- [42] Altares-López S, Ribeiro A and García-Ripoll J J 2021 *Quantum Science and Technology* **6** 045015
- [43] Alotaibi Y 2022 *Symmetry* **14** 623
- [44] Zhang X, Guo A, Ai Y, Tian B and Chen L 2022 *IEEE Transactions on Intelligent Vehicles* **7** 466–79
- [45] Suganthi S, Umamathi N, Mahdal M and Ramachandran M 2022 *Sensors* **22** 1736
- [46] Bowles J, Ahmed S and Schuld M 2024 arXiv:2403.07059