Minimum Accuracy as a Certified Lower Bound for Quantum Kernel-Based Classifiers

Andrias M. M. Cordeiro, Demerson N. Gonçalves, Pedro H. G. Lugão, Tharso D. Fernandes and João T. Dias

***Abstract*— The *minimum accuracy* heuristic, introduced by Suzuki et al., estimates a lower bound on the training accuracy of quantum kernel-based classifiers without explicit model optimization. Its computation, however, relies on the full Pauli decomposition of the feature map, which scales as 4^n for n qubits, making it impractical for large systems. In this work, we prove that minimum accuracy is indeed a certified lower bound on the optimal classification accuracy, and we extend the formulation to arbitrary datasets, without restrictions on size or class balance. Although the method remains computationally expensive for many qubits, our results establish its theoretical validity and broaden its applicability as a useful auxiliary tool for evaluating quantum feature maps and guiding model selection in quantum support vector machines (QSVMs).**

***Keywords*— Quantum kernel methods, Minimum accuracy, Feature map evaluation, Pauli decomposition.**

I. Introduction

Machine learning (ML) is a key part of modern science and technology. It is used for tasks such as recognizing patterns in data and making decisions when there is uncertainty. Quantum machine learning (QML) is a growing field that aims to go beyond classical ML by using the rich structure of quantum systems. One of the most promising QML approaches is based on quantum kernel methods. In these methods, classical data is mapped to quantum states through parameterized quantum circuits, and classical algorithms like support vector machines (SVMs) use these quantum states to perform classification in potentially more powerful feature spaces.

Quantum kernel methods are well suited to Noisy Intermediate-Scale Quantum (NISQ) devices. The expensive part of the computation, which is calculating the kernel, is handled by the quantum computer, while the optimization and training are done on a classical computer. Important works such as those by Havlíček et al. [1] and Schuld and Killoran [2] introduced the idea of quantum feature maps, forming the basis for this hybrid quantum classical framework. Later research has studied how expressive these maps can be, how well they generalize, and how to implement them on real quantum hardware [3], [4], [5].

A practical challenge is how to compare different quantum feature maps without having to train a complete QSVM for each one. Suzuki et al. [6] proposed a heuristic called *minimum accuracy*, which estimates the best possible classification accuracy when using only axis-aligned Pauli measurements. This approach avoids any optimization step and can be resource-efficient for small and medium-scale quantum systems, since it eliminates the need for repeated training. However, computing minimum accuracy requires a full Pauli decomposition, which scales as 4^n for n qubits, making it impractical for large systems.

In this work we give a solid mathematical foundation for minimum accuracy by proving that it is always a valid lower bound for the best classification accuracy achievable by a quantum kernel-based classifier. We also generalize the original definition so that it works for any dataset size, whether balanced or unbalanced. Our results do not make the method more efficient for large n, but they give it a clear theoretical meaning and extend its range of applicability. This makes minimum accuracy a useful supporting tool for evaluating quantum feature maps and guiding model selection in QSVMs, especially in small and medium scale quantum systems.

The rest of this paper is organized as follows. Section II introduces the generalized minimum accuracy framework, including the Pauli representation, formal definition of $R_{\min}$, and an illustrative example. Section III proves that $R_{\min} \leq R^*$, establishing it as a certified lower bound on optimal SVM accuracy. Section IV discusses computational trade-offs, limitations, and potential scalability improvements. Section V concludes with a summary and possible extensions to multiclass and regression tasks.

II. Generalized Minimum Accuracy for Binary Quantum Classification

We consider the problem of binary classification in the quantum kernel framework. The goal is to assign a label $y \in \{+1, -1\}$ to an input vector $\mathbf{x} \in \mathcal{X} \subseteq \mathbb{R}^m$, given a dataset $\{(\mathbf{x}_k, y_k)\}_{k=1}^N$ with N labeled examples.

In quantum kernel methods, the input $\mathbf{x}$ is encoded into an n-qubit quantum state by a *quantum feature map*:

$$|\Phi(\mathbf{x})\rangle = U(\mathbf{x})|0\rangle^{\otimes n}, \tag{1}$$

where $U(\mathbf{x})$ is a parameterized quantum circuit. The similarity between two encoded inputs $\mathbf{x}$ and $\mathbf{z}$ is measured by the kernel

$$K(\mathbf{x}, \mathbf{z}) = |\langle \Phi(\mathbf{x}) | \Phi(\mathbf{z}) \rangle|^2, \tag{2}$$

which corresponds to the fidelity between their quantum states. This kernel can then be used by a classical support vector machine (SVM) to build the decision boundary.

Andrias M. M. Cordeiro, Dep. Eng. Computação, CEFET-RJ, Petrópolis-RJ, andrias.cordeiro@aluno.cefet-rj.br; Demerson N. Gonçalves, Dep. de Matemática, CEFET-RJ, Petrópolis-RJ, demerson.goncalves@cefet-rj.br; Pedro H. G. LUGAO, Dep. Eng. Computação, CEFET-RJ, Petrópolis-RJ, pedro.lugao@cefet-rj.br; João T. Dias, Dep. Eng. Telecomunicações, CEFET-RJ, Rio de Janeiro-RJ, joao.dias@cefet-rj.br .

A. Pauli representation of quantum states

To study and compare quantum feature maps, we represent quantum states in the Pauli operator basis. Let $\mathcal{P}_n = \{I, X, Y, Z\}^{\otimes n}$ be the set of all n-qubit Pauli strings. This set has 4^n elements and forms an orthonormal basis for the space of Hermitian operators on $\mathcal{H} \cong \mathbb{C}^{2^n}$ with respect to the Hilbert–Schmidt inner product.

Any pure state density matrix $\rho(\mathbf{x}) = |\Phi(\mathbf{x})\rangle\langle\Phi(\mathbf{x})|$ can be written as

$$\rho(\mathbf{x}) = \frac{1}{2^n} \sum_{i=1}^{4^n} a_i(\mathbf{x}) \sigma^i, \tag{3}$$

where $\sigma^i \in \mathcal{P}_n$ and

$$a_i(\mathbf{x}) = \mathrm{Tr}[\rho(\mathbf{x})\, \sigma^i] \in \mathbb{R} \tag{4}$$

is the expectation value of σ^i for $|\Phi(\mathbf{x})\rangle$. Each $a_i(\mathbf{x})$ can be seen as a real-valued feature extracted from the quantum state. Hence, a data point $\mathbf{x}$ is encoded in a real feature space $\mathbb{R}^{4^n}$ by the vector

$$\mathbf{a}(\mathbf{x}) = [a_1(\mathbf{x}), a_2(\mathbf{x}), \ldots, a_{4^n}(\mathbf{x})]. \tag{5}$$

The vector $\mathbf{a}(\mathbf{x})$ is often referred to as a *generalized Bloch vector* because the set of Pauli expectation values provides a complete real parametrization of the quantum state. For $n = 1$, the three nontrivial components of $\mathbf{a}(\mathbf{x})$ coincide exactly with the usual Bloch vector representation on the Bloch sphere. For $n > 1$, $\mathbf{a}(\mathbf{x})$ lies in a highly constrained convex subset of $\mathbb{R}^{4^n-1}$ determined by the positivity and normalization of ρ. Although its geometry is more complex than a sphere, the parametrization remains complete.

This representation also allows the quantum kernel

$$K(\mathbf{x}, \mathbf{z}) = |\langle\Phi(\mathbf{x})|\Phi(\mathbf{z})\rangle|^2 \tag{6}$$

to be expressed as a scaled inner product between generalized Bloch vectors. Using $\rho(\mathbf{x}) = \frac{1}{2^n}\sum_{i=1}^{4^n} a_i(\mathbf{x})\sigma^i$ and the orthogonality $\mathrm{tr}(\sigma^i\sigma^j) = 2^n\delta_{ij}$, we find:

$$K(\mathbf{x}, \mathbf{z}) = \mathrm{tr}[\rho(\mathbf{x})\rho(\mathbf{z})] = \frac{1}{2^n} \sum_{i=1}^{4^n} a_i(\mathbf{x}) a_i(\mathbf{z}). \tag{7}$$

The factor $1/2^n$ comes from our choice of using the *unnormalized* Pauli basis, where $\mathrm{tr}(\sigma^i\sigma^j) = 2^n\delta_{ij}$. If instead we normalize each Pauli string as $\tilde{\sigma}^i = \sigma^i/\sqrt{2^n}$, the kernel takes the form $K(\mathbf{x}, \mathbf{z}) = \sum_i \tilde{a}_i(\mathbf{x})\tilde{a}_i(\mathbf{z})$ without the prefactor.

The real vector $\mathbf{a}(\mathbf{x})$ introduced above captures all physically relevant information about the encoded quantum state, with each coordinate corresponding to the expectation value of a distinct Pauli string. This means the original complex Hilbert space $\mathcal{H} \cong \mathbb{C}^{2^n}$ can be equivalently represented as a real feature space $\mathbb{R}^{4^n}$ whose axes are the Pauli strings themselves. In this representation, finding good classification boundaries reduces to searching for separations along these Pauli-feature axes, a viewpoint that underlies the generalized minimum accuracy method described in the next subsection.

B. Generalized minimum accuracy

We consider a binary classification problem with a dataset $D = \{(\mathbf{x}_k, y_k) \mid k = 1, \ldots, N\}$, where each label y_k takes values in $\{+1, -1\}$. A quantum feature map Φ encodes each classical data point $\mathbf{x}_k$ into a quantum state $|\Phi(\mathbf{x}_k)\rangle$. As shown in the previous subsection, each state can be represented by its *generalized Bloch vector*

$$\mathbf{a}(\mathbf{x}_k) = \big[a_1(\mathbf{x}_k), a_2(\mathbf{x}_k), \ldots, a_{4^n}(\mathbf{x}_k)\big] \in \mathbb{R}^{4^n},$$

where $a_i(\mathbf{x}_k)$ is the expectation value of the Pauli string $\sigma^i \in \mathcal{P}_n$. We call each coordinate direction i in $\mathbb{R}^{4^n}$ a *Pauli-feature axis*.

The *minimum accuracy* method evaluates, for each Pauli-feature axis i, how well the dataset can be separated by applying a threshold on that coordinate. For a given axis i, a *threshold* $\tau \in \mathbb{R}$ is a scalar value used to split the dataset into two groups: one containing all samples whose projection $a_i(\mathbf{x}_k)$ is less than τ, and the other containing all samples whose projection is greater than or equal to τ.

We extend the definition of Suzuki et al. [6] so that it works for any dataset size N and for unbalanced class distributions. Let:

- N_+^{total}: total number of samples with label $+1$,
- N_-^{total}: total number of samples with label -1,
- N_+^τ: number of $+1$ samples with $a_i(\mathbf{x}_k) < \tau$,
- N_-^τ: number of -1 samples with $a_i(\mathbf{x}_k) < \tau$.

For a given Pauli-feature axis i and threshold τ, the classification accuracy is:

$$R_i^\tau = \frac{1}{N} \max\left\{N_+^\tau + (N_-^{\mathrm{total}} - N_-^\tau), N_-^\tau + (N_+^{\mathrm{total}} - N_+^\tau)\right\}. \tag{8}$$

This assigns the majority class to each side of the threshold to maximize correct predictions. Then, the *generalized minimum accuracy* is defined by:

$$R_{\min} = \max_{i \in 1,\ldots,4^n} \max_{\tau \in \mathbb{R}} R_i^\tau, \tag{9}$$

where the outer maximization searches over all Pauli-feature axes and the inner maximization searches over all possible thresholds along that axis.

In practice, only $N+1$ candidate thresholds need to be tested for each axis: one before the smallest projection, one after the largest projection, and one between each pair of consecutive sorted projection values $a_i(\mathbf{x}_k)$.

This definition reduces to Suzuki's original formula when the dataset is perfectly balanced, but it remains valid for any N and any class distribution. Geometrically, $R_{\min}$ corresponds to the best separation achievable within the restricted class of axis-aligned hyperplanes in the Pauli-feature space $\mathbb{R}^{4^n}$.

The computation of $R_{\min}$ can be implemented by sorting the projections along each Pauli-feature axis and testing only $N+1$ candidate thresholds. Algorithm 1 summarizes the procedure.

C. Generalized minimum accuracy on a single Pauli-feature axis

This example walks through the computation of the generalized minimum accuracy on one fixed Pauli-feature axis. The

Algorithm 1 Generalized Minimum Accuracy $R_{\min}$

Require: Dataset $\mathcal{D} = \{(\mathbf{x}_k, y_k)\}_{k=1}^{N}$, labels $y_k \in \{+1, -1\}$
Require: Pauli feature projections $a_i(\mathbf{x}_k)$ for $i = 1, \ldots, 4^n$
Ensure: $R_{\min}$ (generalized minimum accuracy)

function COMPUTERMIN($\mathcal{D}, \{a_i(\mathbf{x}_k)\}$)
  $N_+^{\text{total}} \leftarrow |\{k \mid y_k = +1\}|$ ▷ Count +1 class samples
  $N_-^{\text{total}} \leftarrow |\{k \mid y_k = -1\}|$ ▷ Count -1 class samples
  $R_{\min} \leftarrow 0$
  for $i \leftarrow 1$ **to** 4^n **do** ▷ Loop over all Pauli features
    $\mathcal{L} \leftarrow [\,]$ ▷ Initialize empty list
    for $k \leftarrow 1$ **to** N **do**
      Append $(a_i(\mathbf{x}_k), y_k)$ to $\mathcal{L}$
    end for
    Sort $\mathcal{L}$ by $a_i(\mathbf{x}_k)$ in ascending order
    $P_+[0] \leftarrow 0$ ▷ Initialize prefix sums
    $P_-[0] \leftarrow 0$
    for $t \leftarrow 1$ **to** N **do**
      if $\mathcal{L}[t].y = +1$ **then**
        $P_+[t] \leftarrow P_+[t-1] + 1$
        $P_-[t] \leftarrow P_-[t-1]$
      else
        $P_+[t] \leftarrow P_+[t-1]$
        $P_-[t] \leftarrow P_-[t-1] + 1$
      end if
    end for
    for $t \leftarrow 0$ **to** N **do** ▷ Evaluate all thresholds
      $N_+^{\text{left}} \leftarrow P_+[t]$
      $N_-^{\text{left}} \leftarrow P_-[t]$
      $N_+^{\text{right}} \leftarrow N_+^{\text{total}} - N_+^{\text{left}}$
      $N_-^{\text{right}} \leftarrow N_-^{\text{total}} - N_-^{\text{left}}$
      $\text{acc}_1 \leftarrow N_+^{\text{left}} + N_-^{\text{right}}$ ▷ Left: +1, Right: -1
      $\text{acc}_2 \leftarrow N_-^{\text{left}} + N_+^{\text{right}}$ ▷ Left: -1, Right: +1
      $R_{i,\tau} \leftarrow \max(\text{acc}_1, \text{acc}_2)/N$
      if $R_{i,\tau} > R_{\min}$ **then**
        $R_{\min} \leftarrow R_{i,\tau}$ ▷ Update global maximum
      end if
    end for
  end for
  return $R_{\min}$
end function

goal is to make the mechanics of the algorithm transparent. We therefore use a small, synthetic dataset that is *odd-sized* and *class-imbalanced*, highlighting the departure from the balanced, even-N setting in [6].

We observe $N = 11$ samples already *sorted* by their projection on axis i:

$a_i(\mathbf{x}_k)$	0.2	0.5	0.7	1.0	1.2	1.3	1.5	1.8	2.0	2.5	2.7
y_k	-1	-1	-1	+1	-1	-1	-1	-1	+1	+1	+1

TABELA I: Values of $a_i(\mathbf{x}_k)$ and associated class labels y_k.

with class totals $N_+^{\text{total}} = 4$ and $N_-^{\text{total}} = 7$. For any threshold τ placed between two consecutive projection values (and at the two extremes), the line splits into a *left* region ($a_i < \tau$) and a *right* region ($a_i \geq \tau$). We count positives and negatives on each side and evaluate the two possible assignments left as $+1$ and right as -1, or the reverse, keeping the better of the two. The accuracy at τ is the number of correct labels divided by N. The best value over all candidate τ on this axis is denoted by R_i^*.

To fix ideas, we *repeat the procedure twice* for two representative thresholds:

Threshold between 0.2 and 0.5 The left side contains only the sample at 0.2 with label -1. The right side contains the remaining ten samples, with four positives and six negatives. Assigning left as $+1$ and right as -1 yields six correct labels; assigning left as -1 and right as $+1$ yields five. The better assignment therefore achieves $6/11 \approx 0.5455$.

Threshold between 1.3 and 1.5 The left side now contains the first six samples, with labels $[-1, -1, -1, +1, -1, -1]$, i.e., one positive and five negatives. The right side contains the last five samples with labels $[-1, -1, +1, +1, +1]$, i.e., three positives and two negatives. Assigning left as $+1$ and right as -1 gives three correct labels; assigning left as -1 and right as $+1$ gives eight. The better assignment thus achieves $8/11 \approx 0.7273$.

Table II shows the evaluation of all possible thresholds for this axis. Each row reports the candidate threshold interval, the chosen class for the left and right groups, the resulting classification accuracy, and the distribution of samples on each side.

Threshold	Left Class	Right Class	Accuracy	Samples Left	Samples Right
< 0.2	L:+1	R:-1	0.636	0 (0+/0-)	11 (4+/7-)
$[0.2, 0.5)$	L:+1	R:-1	0.545	1 (0+/1-)	10 (4+/6-)
$[0.5, 0.7)$	L:-1	R:+1	0.545	2 (0+/2-)	9 (4+/5-)
$[0.7, 1.)$	L:-1	R:+1	0.636	3 (0+/3-)	8 (4+/4-)
$[1, 1.2)$	L:-1	R:+1	0.545	4 (1+/3-)	7 (3+/4-)
$[1.2, 1.3)$	L:-1	R:+1	0.636	5 (1+/4-)	6 (3+/3-)
$[1.3, 1.5)$	L:-1	R:+1	0.727	6 (1+/5-)	5 (3+/2-)
$[1.5, 1.8)$	L:-1	R:+1	0.818	7 (1+/6-)	4 (3+/1-)
$[\mathbf{1.8}, \mathbf{2.})$	L:-1	R:+1	**0.909**	8 (1+/7-)	3 (3+/0-)
$[2., 2.5)$	L:-1	R:+1	0.818	9 (2+/7-)	2 (2+/0-)
$[2.5, 2.7)$	L:-1	R:+1	0.727	10 (3+/7-)	1 (1+/0-)
> 2.7	L:-1	R:+1	0.636	11 (4+/7-)	0 (0+/0-)

TABELA II: Evaluation of all possible thresholds for the example dataset along a single Pauli-feature axis.

From this exhaustive evaluation, the optimal threshold is found between 1.8 and 2.0, with $\tau \approx 1.9$, achieving an accuracy of $R_{\min} = 0.909$ (90.9%). In this configuration, all samples with $a_i < \tau$ are assigned to class -1 and all with $a_i \geq \tau$ to class $+1$, resulting in only one misclassification.

In the full method, this per-axis maximization is performed for every Pauli-feature coordinate, and the reported value is

$$R_{\min} = \max_{i \in \{1 \ldots 4^n\}} R_i^*.$$

III. MINIMUM ACCURACY AS A LOWER BOUND ON SVM ACCURACY

In this section, we relate the minimum accuracy method to the SVM framework in a fixed feature space $\mathbb{R}^d$ induced by a feature map Φ, which may itself arise from any kernel function

(classical or quantum). Regardless of whether the kernel is linear or non-linear in the input space $\mathcal{X} \subset \mathbb{R}^m$, the SVM is linear in $\mathbb{R}^d$. In typical applications, and especially in the quantum setting where $d = 4^n$, the feature-space dimension d can be much larger than the input dimension m ($d \gg m$). Our result therefore applies to any SVM operating in this fixed high-dimensional feature space.

Let $\Phi : \mathcal{X} \to \mathbb{R}^d$ be a fixed feature map, where in the quantum case $d = 4^n$ and

$$\Phi(\mathbf{x}) \equiv \mathbf{a}(\mathbf{x}) = \big[a_1(\mathbf{x}), \ldots, a_{4^n}(\mathbf{x})\big],$$

with $a_i(\mathbf{x}) = \mathrm{Tr}\big[\rho(\mathbf{x})\sigma^i\big]$ denoting the expectation value of the i-th Pauli string for the quantum state $\rho(\mathbf{x})$ encoding $\mathbf{x}$. Within the classical SVM framework, the hypothesis class of linear classifiers in this feature space is the set $\mathcal{F}$ of all functions $f_{w,b} : \mathcal{X} \to \{-1, +1\}$ of the form

$$f_{w,b}(\mathbf{x}) = \mathrm{sign}(\langle \mathbf{w}, \Phi(\mathbf{x})\rangle + b)\,, \tag{10}$$

where $\mathbf{w} \in \mathbb{R}^d$ is the weight vector and $b \in \mathbb{R}$ is the bias term. Although the kernel function used to construct Φ may be non-linear in the original input space $\mathcal{X} \subset \mathbb{R}^m$ (with $d \gg m$ in typical applications), the SVM decision function remains linear in the feature space $\mathbb{R}^d$.

The empirical accuracy of a classifier $f \in \mathcal{F}$ on a dataset $D = \{(\mathbf{x}_k, y_k)\}_{k=1}^N$, where $y_k \in \{-1, +1\}$, is given by

$$R(f) = \frac{1}{N}\sum_{k=1}^{N} \mathbf{1}[f(\mathbf{x}_k) = y_k]\,, \tag{11}$$

with $\mathbf{1}[\cdot]$ denoting the indicator function, which equals 1 if its argument is true and 0 otherwise. Thus, $\mathbf{1}[f(\mathbf{x}_k) = y_k]$ contributes 1 when the prediction matches the true label and 0 when it does not, making $R(f)$ the proportion of correctly classified samples in D.

The minimum accuracy method restricts $\mathcal{F}$ to a much smaller subclass, namely those classifiers that depend on a *single* Pauli-feature coordinate. In this case, an *axis-aligned classifier* has the form

$$f_{i,\tau}(\mathbf{x}) = \mathrm{sign}\big(a_i(\mathbf{x}) - \tau\big), \tag{12}$$

with $i \in \{1, \ldots, 4^n\}$ and $\tau \in \mathbb{R}$. We denote by $\mathcal{F}_{\mathrm{axis}}$ the set of all such classifiers.

Geometrically, each $f_{i,\tau}$ corresponds to a hyperplane in $\mathbb{R}^{4^n}$ orthogonal to the i-th coordinate axis, with a weight vector of the form $w = (0, \ldots, 0, w_i, 0, \ldots, 0)$ with $w_i \neq 0$. Since every axis-aligned classifier is, in particular, a valid linear classifier in $\mathcal{F}$, we have the strict inclusion

$$\mathcal{F}_{\mathrm{axis}} \subseteq \mathcal{F}. \tag{13}$$

It follows that

$$\sup_{f \in \mathcal{F}_{\mathrm{axis}}} R(f) \;\leq\; \sup_{f \in \mathcal{F}} R(f). \tag{14}$$

By definition, the left-hand side is precisely the *generalized minimum accuracy*:

$$R_{\min} := \sup_{f \in \mathcal{F}_{\mathrm{axis}}} R(f), \tag{15}$$

while the right-hand side corresponds to the optimal empirical accuracy achievable by a fully optimized linear SVM in the same feature space:

$$R^* := \sup_{f \in \mathcal{F}} R(f). \tag{16}$$

Thus, we obtain the inequality

$$R_{\min} \leq R^*. \tag{17}$$

The argument presented above can be formally stated as the following theorem, which establishes the minimum accuracy as a certified lower bound on SVM performance in the same feature space.

Theorem 1 (Minimum accuracy lower bound): Let $\mathcal{F}$ be the hypothesis class of all linear classifiers in the feature space induced by Φ, and let $\mathcal{F}_{\mathrm{axis}} \subset \mathcal{F}$ be the subclass of axis-aligned classifiers. Then, for any fixed dataset,

$$R_{\min} := \sup_{f \in \mathcal{F}_{\mathrm{axis}}} R(f) \;\leq\; \sup_{f \in \mathcal{F}} R(f) =: R^*,$$

where $R(f)$ is the empirical classification accuracy of f.

This result shows that $R_{\min}$ can be interpreted as the best accuracy achievable under a severe structural restriction on the decision boundary, namely, being orthogonal to a single Pauli-feature axis. Since this restriction is lifted in the full SVM optimization, $R_{\min}$ serves as a provable lower bound for R^*, providing a computationally efficient, non-optimized benchmark for quantum feature maps in $\mathbb{R}^{4^n}$.

IV. Discussions and Future Works

The results presented in this work position the minimum accuracy metric $R_{\min}$ as a practical and theoretically grounded tool for evaluating quantum feature maps. Computed entirely on classical hardware from Pauli feature vectors $\mathbf{a}(\mathbf{x}) \in \mathbb{R}^{4^n}$, $R_{\min}$ serves as a certified lower bound on the maximum empirical accuracy achievable by any SVM operating in the same feature space. This bound holds irrespective of the underlying kernel function, as the SVM decision surface remains linear in the induced feature space $\mathbb{R}^d$.

From a computational perspective, the advantage of the proposed method becomes even more evident when compared to established baselines. Its overall complexity, $O(4^n N \log N)$, arises from sorting and evaluating axis-aligned thresholds, making it substantially more efficient than the $O(N^2 \cdot 4^n + N^3)$ cost of training a kernelized SVM with explicit features [7]. In the quantum setting, the authors in [8] show that dual-formulation QSVM training demands $\tilde{O}(N^{4.67}/\varepsilon^2)$ quantum circuit evaluations, where N is the number of samples and ε is the target accuracy relative to the optimum with exact expectation values. The absence of iterative optimization steps allows $R_{\min}$ to serve as an efficient pre-screening filter for feature maps: configurations with very low $R_{\min}$ (e.g., close to 50%) can be discarded early, thereby avoiding unnecessary quantum evaluations. This capability is particularly valuable in NISQ-era workflows, where quantum processing time remains both scarce and costly.

The approach is not without limitations. Its reliance on the explicit Pauli decomposition can become computationally

prohibitive for $n > 5$ qubits due to exponential scaling. Moreover, $R_{\min}$ is a conservative estimate; while a high value (e.g., above 90%) is strongly indicative of separability, a low value does not necessarily imply poor QSVM performance, since the optimal separating hyperplane may combine multiple Pauli axes in ways the axis-aligned restriction cannot capture. The method therefore complements, rather than replaces, more expressive training procedures.

Compared directly to classical SVMs in the same feature space $\mathbb{R}^{4^n}$, $R_{\min}$ retains a decisive advantage for $n \leq 4$ and moderate dataset sizes ($N \leq 10^4$), as the $N \log N$ scaling is far more favorable than the cubic dependence on N characteristic of dense quadratic programming. Beyond $n = 5$, both approaches face intractable resource demands, motivating the exploration of approximate strategies.

Future research should address this scalability bottleneck. A natural direction is to estimate $R_{\min}$ through Monte Carlo sampling over a subset of Pauli axes, providing probabilistic guarantees without enumerating the full 4^n set. Another possibility is to exploit dimensionality-reduction methods capable of compressing Pauli feature vectors $\mathbf{a}(\mathbf{x})$ into a lower-dimensional representation while retaining their discriminative structure. These approaches could allow the method to scale beyond current limits, enabling its use as a practical screening tool for larger quantum systems.

Also, the conceptual framework of minimum accuracy could be extended beyond binary classification. Generalizations to multiclass classification and regression would require adapting the axis-aligned decision rule, potentially leading to new quantum–classical hybrid models. An open question is whether high values of $R_{\min}$ correlate with quantum-induced separability that cannot be reproduced by classical kernels, making it a potential diagnostic tool for identifying scenarios of genuine quantum advantage.

Since one of the main contributions of this work was to provide a formal proof that the minimum accuracy constitutes a lower bound on the optimal SVM accuracy in the same feature space, it is also worth exploring whether this mathematical framework can be leveraged to derive other theoretical guarantees. In particular, analogous lower bounds could potentially be formulated for regression problems, where mean squared error (MSE) or mean absolute error (MAE) would replace accuracy as the primary metric. More broadly, this line of reasoning could be extended to other quantum machine learning paradigms, such as quantum regression, quantum anomaly detection, or kernel-based dimensionality reduction, where a formally proven baseline performance could guide both algorithm design and feature map selection.

V. Conclusion

This work establishes a rigorous theoretical foundation for the minimum accuracy metric ($R_{\min}$) in quantum kernel-based classifiers. We formally prove that $R_{\min}$ constitutes a certified lower bound on the optimal accuracy achievable by a SVM in the same quantum feature space (i.e., $R_{\min} \leq R^*$). Additionally, we generalize the original definition to accommodate arbitrary datasets (any size N and imbalanced class distributions), broadening its practical scope.

Computationally, $R_{\min}$ offers a more efficient classical pre-screening tool ($O(4^n N \log N)$) for evaluating quantum feature maps. Its ability to identify promising embeddings (e.g., high $R_{\min}$) avoids costly quantum optimizations in the NISQ era. However, dependence on the full Pauli decomposition (4^n terms) limits scalability beyond $n > 5$ qubits.

Future work must address this exponential barrier via Pauli-axis sampling, feature compression, or extensions to multiclass/regression. The proven lower-bound property also enables exploration of analogous guarantees for other quantum learning paradigms.

Acknowledgments

Andrias M. M. Cordeiro acknowledges financial support from CEFET/RJ through the Institutional Program of Scientific Initiation Scholarships (PIBIC/2024) for Undergraduates under the Research Grant Call.

Referências

[1] V. Havlíček, A. D. Córcoles, K. Temme, A. W. Harrow, A. Kandala, J. M. Chow e J. M. Gambetta, "Supervised learning with quantum-enhanced feature spaces," *Nature*, vol. 567, no. 7747, pp. 209–212, 2019. doi: 10.1038/s41586-019-0980-2.

[2] M. Schuld e N. Killoran, "Quantum machine learning in feature Hilbert spaces," *Physical Review Letters*, vol. 122, no. 4, p. 040504, 2019. doi: 10.1103/PhysRevLett.122.040504.

[3] F. Tacchino, C. Macchiavello, D. Gerace e D. Bajoni, "Quantum machine learning: a classical perspective," *npj Quantum Information*, vol. 5, no. 1, pp. 1–8, 2019. doi: 10.1038/s41534-019-0157-8.

[4] H.-Y. Huang, M. Broughton, M. Mohseni, R. Babbush, R. Kueng, J. Preskill e J. R. McClean, "Power of data in quantum machine learning," *Nature Communications*, vol. 12, no. 1, p. 2631, 2021. doi: 10.1038/s41467-021-22539-9.

[5] M.-C. Li, J.-G. Liu, X.-Y. Pan e H. Fan, "Quantum kernel evaluation on NISQ devices with random feature maps," *Quantum Science and Technology*, vol. 7, no. 4, p. 045025, 2022. doi: 10.1088/2058-9565/ac8c56.

[6] Y. Suzuki, H. Yano, Q. Gao, S. Uno, T. Tanaka, M. Akiyama e N. Yamamoto, "Analysis and synthesis of feature map for kernel-based quantum classifier," *Quantum Machine Intelligence*, vol. 4, no. 1, pp. 1–15, 2022. doi: 10.1007/s42484-022-00058-8.

[7] A. Bordes, S. Ertekin, J. Weston e L. Bottou, "Fast kernel classifiers with online and active learning," *Journal of Machine Learning Research*, vol. 6, no. 54, pp. 1579–1619, 2005.

[8] G. Gentinetta, A. Thomsen, D. Sutter e S. Woerner, "The complexity of quantum support vector machines," *Quantum*, vol. 8, p. 1225, 2024.

[9] J. Wakeham e M. Schuld, "Inference, interference and invariance: How the Quantum Fourier Transform can help to learn from data," *Quantum*, vol. 5, p. 515, 2024.

[10] L. Banchi, S. Severini e M. M. Wilde, "Generalization in quantum machine learning: A quantum information standpoint," *PRX Quantum*, vol. 2, no. 4, p. 040321, 2021. doi: 10.1103/PRXQuantum.2.040321.