Quantum Classifiers applied to Multi-User Detector in Communications Systems

João T. Dias, Demerson N. Gonçalves, Felipe A. Silva and Daniel C. Neves

***Abstract*— In this paper we implement two quantum machine learning (QML) algorithms, namely, quantum kernel-based classifier (QKC) and variational quantum classifier (VQC), considering different types of encoding functions. We compare its performance to that of its classical counterpart, the classical support vector machine (SVM) applied to the detector in a multi-user - direct sequence - code division multiple access (MU-DS-CDMA) system, with a scenario where the user code generates non-linearities as a function of the channel delay profile. The results explicitly prove accuracy advantage achieved by our quantum classifiers on three types of datasets. This work shows that quantum classifiers have a huge potential to be useful in the future as the number of qubits in the quantum computer increases.**

***Keywords*— Quantum classifiers; MU-DS-CDMA; Multi-User Detection.**

I. Introduction

High spectral efficiency, massive connectivity and low latency are among the requirements for next-generation communications systems, and these requirements are expected to increase as researchers focus their efforts on sixth-generation (6G) wireless communications. Massive multiple-input multiple-output (M-MIMO), non-orthogonal multiple access (NOMA) and millimeter wave (mmWave) communications are promising techniques to meet these stringent requirements [1].

Supporting a large number of users communicating over a common channel may not be readily achievable by orthogonal multiple access (OMA) systems due to multiple access interference (MAI). It is widely known that the complexity of an optimal detector is exponentially proportional to the number of users, which prevents its practical implementation. Several sub optimal low-complexity detection techniques have already been proposed [2].

Recently, several NOMA solutions have been actively investigated [1], which can basically be divided into two main categories, namely, power-domain NOMA and code-domain NOMA. Some of NOMA's strong competitors in the code domain are low-density spreading-aided CDMA (LDS-CDMA) [3] and sparse code multiple access (SCMA) [4].

In parallel with the development of telecommunications systems, quantum computing has been developing and proving to be capable of solving problems considered prohibitive due to its computational complexity [5], [6]. Quantum computing is a computational paradigm based on the laws of quantum mechanics, which promises to solve problems that would take classical computing too long to compute due to the size of the data set or the computational power required. In QML, many quantum algorithms are developed by adapting classical algorithms or their expensive subroutines to run on a potential quantum computer. There are many algorithms of QML such as solving linear systems of equations, principal component analysis (QPCA) and support vector machines [7].

This work is motivated to show that a solution to the problem of an optimal detector complexity can be obtained by using quantum algorithms. For this, some QML algorithms such as QKC and VQC were studied and the performance of a multi-user detector based on QML in a MU-DS-CDMA system was analyzed with a scenario where the user code generates non-linearities as a function of the delay profile of the channel. The results showed the viability of the proposed solution and contribute to a new approach in the quest to meet the requirements of future communication systems.

This work is organized as follows: the Theory Fundamentals are described in section II, The quantum classifiers are presented in section III. In section IV, the simulation results are presented, and conclusions are made in section V.

II. Theory Fundamentals

A. MU-DS-CDMA system model

The model of MU-DS-CDMA used in this work is depicted in Fig. 1 [8].

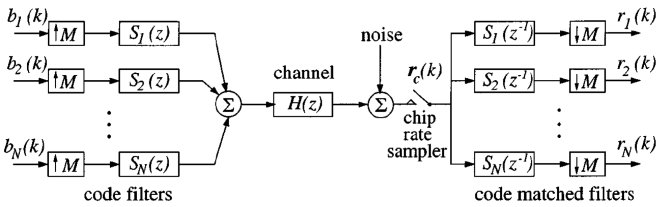

Fig. 1. Structure of MU-DS-CDMA System.

The value $b_i(k) \in \{\pm 1\}$ denotes the kth bit of user i, N is the number of users, M is the length of the user code and $S_i(z) \in \{\pm 1\}$ is the z-th element of the user signature. The received signal $\mathbf{r}(k)$ is given by

$$\mathbf{r}(k) = \mathbf{P}\left[b(k) \ldots b(k-L+1)\right]^T + \bar{\mathbf{S}}^T \mathbf{n}(k), \qquad (1)$$

where $\bar{\mathbf{S}} = [\bar{\mathbf{s}}_1 \cdots \bar{\mathbf{s}}_N]$ is the normalised user signature sequence matrix, $\mathbf{n}$ is the additive white Gaussian noise vector

João T. Dias, Dep. de Engenharia de Telecomunicações, CEFET-RJ, Rio de Janeiro-RJ, e-mail: joao.dias@cefet-rj.br; Demerson N. Gonçalves, Coord. de Licenciatura em Matemática, CEFET-RJ, Petrópolis-RJ, e-mail:demerson.goncalves@cefet-rj.br; Felipe A. Silva , Coord. de Licenciatura em Matemática, CEFET-RJ, Petrópolis-RJ, e-mail: felipe.silva.6@aluno.cefet-rj.br; Daniel C. Neves, Coord. de Licenciatura em Física, CEFET-RJ, Petrópolis-RJ, e-mail:daniel.neves@aluno.cefet-rj.br .

and the $N \times LN$ system matrix $\mathbf{P} = \bar{\mathbf{S}}^T \mathbf{H}$ diag$(\bar{\mathbf{S}}\mathbf{A}, \ldots, \bar{\mathbf{S}}\mathbf{A})$, where $\mathbf{A}$ is the diagonal user signal amplitude matrix and $\mathbf{H}_{M \times LM}$ is the channel impulse response matrix (CIR). The intersymbol interference span L depends on the length of the CIR, n_h, related to the length of the chip sequence M. For $n_h = 1, L = 1$; for $1 < n_h \leq M, L = 2$; for $M < n_h \leq 2M, L = 3$, and so on.

B. SVM detector

Support Vector Machine (SVM) is a supervised Machine Learning (ML) technique that can be used for classification tasks [9]. Let $\{(\mathbf{x}_j, y_j)\}_{j=1}^{C}$ be a linearly separable dataset, where C is the size of the training dataset with data dimension D and $y_j = \pm 1$, i.e., $y_j = 1$ or $y_j = -1$ depending on the class to which $\mathbf{x_j}$ belongs (binary classification). SVM finds the optimum hyperplane with parameters $(\mathbf{w}, \varrho)$ that divides the two classes, i.e., datapoints with labels $y_j = 1$ that satisfies $y_j(\mathbf{w} \cdot \mathbf{x_j} + \varrho) \geq 1$ and datapoints with labels $y_j = -1$ satisfying $y_j(\mathbf{w} \cdot \mathbf{x_j} + \varrho) \leq -1$. In addition, the boundary of the decision regions $\frac{2}{\|\mathbf{w}\|}$ is maximized. Then, the decision function is given by $f(\mathbf{x}) = \text{sign}\,(\mathbf{w}^* \cdot \mathbf{x_j} + \varrho^*)$, where $\mathbf{w}^*$ and ϱ^* are the optimal values obtained from the optimization procedure. In the dual formulation, the goal is to maximizing $\sum_{j=1}^{C} y_j \varphi_j - \frac{1}{2} \sum_{j,c=1}^{C} \varphi_j K(\mathbf{x_j}, \mathbf{x_c}) \varphi_c$, where $\boldsymbol{\varphi}$ are the Lagrange multipliers, subject to the constraints $\sum_{j=1}^{C} \varphi_j = 0$ and $y_j \varphi_j \geq 0$. Hence, the parameters $(\mathbf{w}^*, \varrho^*)$ can be recovered as $\mathbf{w}^* = \sum_{j=1}^{C} \varphi_j^* \mathbf{x}_j$ and $\varrho^* = y_c - \mathbf{w}^* \cdot \mathbf{x}_c$ for an index c corresponding to $\varphi_c^* \neq 0$. The function $K(\mathbf{x_j}, \mathbf{x_c}) = \mathbf{x_j} \cdot \mathbf{x_c}$ is called *kernel function* and can also be defined for a more complicated non-linear dataset [10]. Putting values $\mathbf{w}^*$ and ϱ^* into the decision function f, we obtain a binary classifier for new data $\mathbf{x}$:

$$f(\mathbf{x}) = \text{sign}\left(\sum_{j=1}^{C} \mathbf{w}^* K(\mathbf{x_j}, \mathbf{x}) + \varrho^* \right). \tag{2}$$

Suppose that the receiver MU-DS-CDMA has access to a block of K training samples $\{\mathbf{r}(k), b_i(k)\}_{k=1}^{K}$. The SVM multi-user detector (SVM MUD) can be designed for each user i by applying the standard SVM to set of noisy received signal [11]:

$$f_{\text{SVM}}(k, i) = \sum_{j=1}^{K} \varphi_j^* b_i(j) K(\mathbf{r}(k), \mathbf{r}(j)) + \varrho^*, \tag{3}$$

and making the decision of user i data with $\hat{b}_i(k) = \text{sign}(f_{\text{SVM}}(k, i))$.

C. Fundamentals of quantum computing

Quantum computing is a model of computation that exploits quantum mechanical phenomena to perform high speed parallel computing [12]. Classical information in digital computers is represented by logical binary digits (bits). A logical bit can take the value of 1 or 0 depending on whether the voltage in the wire is High or Low in a logic circuit. In contrast, the smallest unit of information stored in a two-state quantum computer, called a *quantum bit* or *qubit*, is a unit vector in the two-dimensional complex Hilbert space ($\mathbb{C}^2$) for which a particular orthogonal basis $\{|0\rangle, |1\rangle\}$ has been fixed. The quantum state of a qubit can be represented using any chosen orthogonal basis. The most commonly used basis is the *computational basis*, which corresponds to $|0\rangle = [1, 0]^T$ and $|1\rangle = [0, 1]^T$. Unlike the classical bit, a qubit can be in a linear superposition of $|0\rangle$ and $|1\rangle$, that is, $|\psi\rangle = \alpha |0\rangle + \beta |1\rangle = [\alpha, \beta]^T$ where $\alpha, \beta \in \mathbb{C}$ are the *amplitudes* of $|\psi\rangle$ on the computational basis with the constraint $|\alpha|^2 + |\beta|^2 = 1$. When $\alpha = 1$, then $\beta = 0$ and hence $|\psi\rangle = |0\rangle$, which corresponds to the classical bit value 0. Similarly, if $\alpha = 0$, then $\beta = 1$ and $|\psi\rangle = |1\rangle$, which corresponds to the classical bit value 1. In general, when a state of one qubit $|\psi\rangle$ is *measured* with respect to the computational basis, the probability that the measured value is $|0\rangle$ is $|\alpha|^2$ and the probability that the measured value is $|1\rangle$ is $|\beta|^2$.

The state of a quantum computer can be changed by applying unitary operators or quantum gates to its qubits. One of the most widely used single-qubit unitary operator is the Hadamard gate, given by $H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$. One can easily check that $H |0\rangle = \frac{|0\rangle + |1\rangle}{\sqrt{2}}$ and $H |1\rangle = \frac{|0\rangle - |1\rangle}{\sqrt{2}}$. If the input is $|0\rangle$, the Hadamard gate creates a superposition of states with equal weights. Another important set of single-qubit gates is the *Pauli matrices*,

$$X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, Y = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}, Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}. \tag{4}$$

The gate X is the quantum NOT gate because $X |\psi\rangle = \beta |0\rangle + \alpha |1\rangle$. The Z operator is the gate imposing a *phase shift* by π radians, since it flips the sign of the amplitude of the state $|1\rangle$, $Z |\psi\rangle = \alpha |0\rangle - \beta |1\rangle$. The operator Y can be considered as a combination of X and Z gates, because $Y |\psi\rangle = i(-\beta |0\rangle + \alpha |1\rangle)$.

In order to increase the complexity of a quantum system, a most general 1-qubit gate can be used,

$$U(\theta, \phi, \lambda) = \begin{bmatrix} \cos(\frac{\theta}{2}) & -e^{i\lambda} \sin(\frac{\theta}{2}) \\ e^{i\lambda} \sin(\frac{\theta}{2}) & e^{i(\lambda+\phi)} \cos(\frac{\theta}{2}) \end{bmatrix}. \tag{5}$$

For instances, three useful gates obtained from U are

$$U(\theta, \frac{\pi}{2}, -\frac{\pi}{2}) = R_x(\theta) = \begin{bmatrix} \cos(\frac{\theta}{2}) & -i \sin(\frac{\theta}{2}) \\ -i \sin(\frac{\theta}{2}) & \cos(\frac{\theta}{2}) \end{bmatrix}, \tag{6}$$

$$U(\theta, 0, 0) = R_y(\theta) = \begin{bmatrix} \cos(\frac{\theta}{2}) & -\sin(\frac{\theta}{2}) \\ \sin(\frac{\theta}{2}) & \cos(\frac{\theta}{2}) \end{bmatrix}, \tag{7}$$

$$U(0, 0, \lambda) = e^{\frac{i\lambda}{2}} R_z(\lambda) = \begin{bmatrix} 1 & 0 \\ 0 & e^{i\lambda} \end{bmatrix}, \tag{8}$$

where R_x, R_y and R_z are the operators that rotate the Bloch sphere [12] about the x, y, and z-axis, respectively.

To dealing with multiple qubits is necessary to introduce the concept of *tensor product*. Let V and W be complex vectors space of dimensions m and n, respectively. The tensor product $V \otimes W$ is an mn-dimensional vector space. For example, if we have a two-qubit quantum computer and the first qubit is in the state $|0\rangle$ and the second is in the state $|1\rangle$, then the quantum computer is in the state $|0\rangle \otimes$

$|1\rangle = |01\rangle = [0,1,0,0]^T$. The resulting vector is a four-dimensional vector space. The general state $|\psi\rangle$ of a 2-qubit is a superposition $|\psi\rangle = \alpha|00\rangle + \beta|01\rangle + \gamma|10\rangle + \delta|11\rangle$, with the constraint $|\alpha|^2 + |\beta|^2 + |\gamma|^2 + |\delta|^2 = 1$. In general, the quantum state $|\psi\rangle$ of an n-qubit is a superposition of 2^n states $|0\rangle, |1\rangle, \ldots, |2^n - 1\rangle$ (computational basis in decimal notation), $|\psi\rangle = \sum_{i=0}^{2^n-1} \alpha_i |i\rangle$, with the amplitudes α_i constrained to $\sum_{i=0}^{2^n-1} |\alpha_i|^2 = 1$.

Applying the Hadamard gate to the n-qubit state $|0\rangle$ we obtain $H^{\otimes n}|0\rangle = (H|0\rangle)^{\otimes n} = \frac{1}{\sqrt{2^n}} \sum_{i=0}^{2^n-1} |i\rangle$. Thus, this product produces an equally weighted superposition of all computational basis states, when the input is the state $|0\rangle$. This state is useful for applying *quantum parallelism*. Quantum parallelism is one of the most important features of quantum computers that promise to solve problems that are too hard for classical computers to solve in reasonable amount of time.

To conclude this discussion of basics of quantum computing, let us consider a very important operation on 2-qubit system, the *controlled-NOT* or *CNOT* operation. It has two input qubits, the *control* and the *target* qubit, respectively. The target qubit is flipped only if the control qubit is set to 1, that is, $|a, b\rangle \rightarrow |a, a \oplus b\rangle$, where $\oplus$ is addition modulo 2.

III. Quantum Classifiers

A. Quantum kernel-based classifier

In the classical SVM method, a nonlinear dataset is transformed by a function, called *kernel function*, to another dataset in a high-dimensional feature space, where the data become linearly separable. It can be performed using the so-called *kernel trick*, which allows one to evaluate the kernel function without explicitly mapping the data to the high-dimensional feature space [10].

In the quantum scenario, the quantum feature map encodes the dataset from its original low dimensional real space into a high dimensional Hilbert space in a natural way [13]. So, kernel trick is unnecessary in quantum approaches. The potential of using quantum methods comes from the fact that, in general, quantum mapping cannot be efficiently simulated with classical computers [14], [15]. This is a necessary, but not sufficient, condition to obtain quantum advantage using quantum kernels. In the kernel-based approach, the feature map transforms an input dataset to a set of multi-qubit states. Thus, the kernel matrix is constructed by calculating all the inner products of quantum states and then the standard SVM is used for classifying the dataset. As we can see in Sec. IV (Tables I, II and III), different feature maps lead to different kernels which can influence on the classification accuracy, so a careful analysis of the feature map must be done.

There are several techniques to map classical data into quantum data in high dimensional spaces [13], [14], [16]. In general, each n-dimensional classical data $\mathbf{x} \in \mathbb{R}^n$ is encoded into a unitary operator $\mathcal{U}_{\phi(\mathbf{x})}$ through an *encoding function* $\phi(\mathbf{x})$, and it is applied to the initial state $|0\rangle^{\otimes n}$. That is, the classical data $\mathbf{x} \in \mathbb{R}^n$ is mapped to the quantum state $|\phi(\mathbf{x})\rangle = \mathcal{U}_{\phi(\mathbf{x})}|0\rangle^{\otimes n} \in \mathbb{C}^{2^n}$. Then, the quantum kernel is given by the overlap of two data-encoding quantum states, $K(\mathbf{x}, \mathbf{x}') = |\langle\phi(\mathbf{x})|\phi(\mathbf{x}')\rangle|^2$. This quantity can be efficiently estimated by using the *SWAP test* [17]. Now, we can use this kernel as a subroutine in a classical SVM, which yields a hybrid quantum-classical approach. In this case, the separating hyperplane is constructed in a purely classical manner, only the kernel function between the training datapoints is evaluated on the quantum simulation.

B. Quantum feature map

In this work we present a 2-qubit classifier algorithm applied to a multi-user detection system. Our algorithm is based on Ref. [14] and utilizes both QKC and VQC methods. The main idea is to load classical data from the MU-DS-CDMA model (see Fig. 1) into quantum states, through a procedure called *quantum feature map*.

In [14], the authors presented a set of feature maps that is conjectured to be hard to simulate classically and that can be implemented as short-depth circuits on near-term quantum devices. The quantum feature map is defined by

$$\mathcal{U}_{\Phi(\mathbf{x})} = \prod_d U_{\Phi(\mathbf{x})} H^{\otimes n}, \tag{9}$$

where

$$U_{\Phi(\mathbf{x})} = e^{i \sum_{S \in \mathcal{I}} \phi_S(\mathbf{x}) \prod_{k \in S} P_k}, \tag{10}$$

n is the number of qubits that corresponds to the dimension of the datapoints $\mathbf{x}$, which are encoded through the coefficients $\phi_S(\mathbf{x})$. S is a set of qubit indices that describes the connections in the feature map, $\mathcal{I}$ is a set containing all these index sets and $P_k \in \{\mathbb{I}, X, Y, Z\}$. The encoding function is given by

$$\phi_S : \mathbf{x} \mapsto \begin{cases} x_i & \text{if } S = \{i\} \\ (\pi - x_i)(\pi - x_j) & \text{if } S = \{i, j\}. \end{cases} \tag{11}$$

For example, writing $\phi_S(\mathbf{x}) = \phi_S$ and choosing $n = 2$, $\mathcal{I} = \{\{1\}, \{2\}, \{1, 2\}\}$, P_k being the matrix Pauli Z acting on the k-th qubit, the equation (10) becomes

$$U_{\phi(\mathbf{X})} = e^{i(x_1 Z_1 + x_2 Z_2 + (\pi - x_1)(\pi - x_2) Z_1 Z_2)}. \tag{12}$$

The sequence of Pauli matrices Z_1, Z_2 and $Z_1 Z_2$ in Eq. (12) is denoted by $[Z, Z, ZZ]$.

The paper [14] only considered the feature map given by Eq. (12). Here, inspired by [18], we implement different types of encoding functions for our dataset (13)-(17):

$$\phi_1(\mathbf{x}) = x_1, \phi_2(\mathbf{x}) = x_2, \phi_{1,2}(\mathbf{x}) = \pi x_1 x_2; \tag{13}$$

$$\phi_1(\mathbf{x}) = x_1, \phi_2(\mathbf{x}) = x_2, \phi_{1,2}(\mathbf{x}) = \frac{\pi}{2}(1 - x_1)(1 - x_2); \tag{14}$$

$$\phi_1(\mathbf{x}) = x_1, \phi_2(\mathbf{x}) = x_2, \phi_{1,2}(\mathbf{x}) = \exp\left(\frac{|x_1 - x_2|^2}{8/\ln(\pi)}\right); \tag{15}$$

$$\phi_1(\mathbf{x}) = x_1, \phi_2(\mathbf{x}) = x_2, \phi_{1,2}(\mathbf{x}) = \frac{\pi}{3\cos(x_1)\cos(x_2)}; \tag{16}$$

$$\phi_1(\mathbf{x}) = x_1, \phi_2(\mathbf{x}) = x_2, \phi_{1,2}(\mathbf{x}) = \pi\cos(x_1)\cos(x_2). \tag{17}$$

We also consider different Pauli matrices sequences $[P_1, P_2, P_3 P_4]$, with $P_i \in \{\mathbb{I}, X, Y, Z\}$, in our algorithms.

For example, the Fig. 2 shows the circuit of the quantum operator $\mathcal{U}_{\Phi(\mathbf{x})}$, with $n = 2$ qubits, circuit depth $d = 1$, set of Pauli gates $[X, I, ZY]$ and encoding function (15):

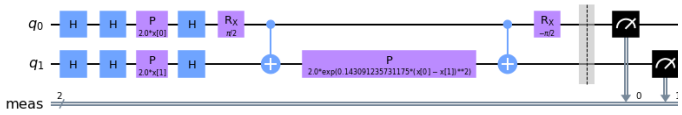

Fig. 2. Circuit of $\mathcal{U}_{\Phi(\mathbf{x})}$ with $n = 2, d = 1$, Pauli gates $[X, I, ZY]$ and encoding function (15).

C. Variational quantum classifier

Variational quantum classifier is a supervised QML algorithm used for classification problems in Noisy Intermediate Scale Quantum (NISQ) Technology [19]. The authors of [14] proposed a VQC model by exploring NISQ devices without the need for extra error correction approaches. VQC depends on free parameters and consists basically in three parts: data encoding (or data loading), a variational circuit (or ansatz) and measurement. The variational quantum circuit is used to encode the data into the quantum circuit and tune the hyper parameters with rotation gates, R_x, R_y and R_z [16], [20]. The cost function measures the similarity between the resultant label of the circuit and the actual label. It also updates the parameters of the variational circuit using iterative device measurements. The optimizer evaluates the cost function with the current parameters and selects the next iteration's parameters until it converges on an optimal solution. The hybrid nature of this family of algorithms comes from the fact that the cost functions are evaluated using quantum resources and optimized through classical ones.

For a set of parameters θ, input data $\mathbf{x}$ with binary output labels $y_i \in \{-1, 1\}$, we will implement a parameterized quantum circuit that outputs the quantum state:

$$|\Psi(\mathbf{x}, \theta)\rangle = U_{W(\theta)} U_{\phi(\mathbf{x})} |0\rangle , \tag{18}$$

where $U_{W(\theta)}$ is the unitary variational circuit and $U_{\phi(\mathbf{x})}$ is the unitary data encoding circuit. In this work, $U_{\phi(\mathbf{x})}$ is implemented using the set of encoding function (11), (13) - (17) and $U_{W(\theta)}$ is implemented using both RealAmplitude and TwoLocal circuits from the Qiskit circuit library [21], as can be shown in Fig. 3 and Fig. 4, respectively.

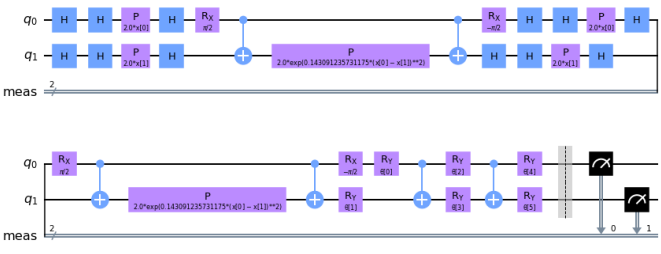

Fig. 3. Ansatz RealAmplitudes with $n = 2, d = 2$, Pauli gates $[X, I, ZY]$ and encoding functions (15).

The VQC model provided by Qiskit library [21] interprets the measured bitstrings as the output of a classifier by calculating the parity of the given bitstring. If the parity is even, it returns a label 1, and if the parity is odd it returns a label -1 [14].

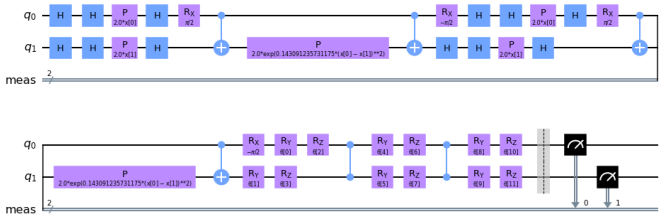

Fig. 4. Ansatz TwoLocal with $n = 2, d = 2$, Pauli gates $[X, I, ZY]$ and encoding functions (15).

IV. Simulations

A. Methods

We used two quantum ML methods for a binary classification problem applied to the MU-DS-CDMA (see Sec. II-A) dataset. These two methods are the QKC and the VQC, briefly described in Sec. III-A and Sec. III-C, respectively. We also utilized the Qiskit library (an open-source quantum computing framework created by IBM®) [21] for the quantum machine learning task with a local quantum simulator. The classical kernel-based method for SVM (Sec. II-B) was run on a classical computer with a regular CPU.

B. Simulation Results

In this section, we present classification accuracy results for different types of quantum kernels and compare to RBF for classical SVM. We demonstrate that the performance of quantum SVM can be comparable to classical SVM and, in some situations, outperform the classical model when kernels becomes more complex. See Tables I, II and III.

We consider a MU-DS-CDMA system supporting $N = 2$ users with $M = 4$ chips per symbol. The chip sequences of the two users were set as (1,1, -1, -1) and (1,-1, -1,1), respectively. The transfer function of the channel impulse responses (CIR) used in this simulation is $H(z) = 0.3 + 0.7z^{-1} + 0.3z^{-2}$ and the additive white Gaussian noise (AWGN) has zero mean and variance σ_n^2. We also consider three signal to noise ratio (SNR) values: 10 dB, 15 dB, and 20 dB. To evaluate the approach, in the time sequence of the received symbols, a set with the first 50 symbols (preamble) is used for training and validation, for each noise condition. In this set we perform a division of 60:40 percent, that is, the first 30 samples are used to train the model and the next 20 for validation.

For each signal noise condition, we performed the tests using the feature maps in Eqs. (11) and (13)-(17) and for each feature map, we used 48 combinations of Paulis matrices of the type $[P_1, P_2, P_3 P_4]$. Thus, for every QKC, VQC RealAmplitude and VQC TwoLocal method, the algorithm was run 6*48=288 times. The tables I, II, III show the accuracy of the best model settings:

We first consider the results for the dataset of 10 dB of SNR. Table I shows that the VQC TwoLocal model with default feature map (11) and Pauli matrices sequence $[Z, I, ZY]$ achieved the best accuracy in terms of train and validation dataset, while the classical RBF outperforms all other methods in the test column. Test represents the accuracy of the model

TABLE I
COMPARISON PERFORMANCE OF SVM DETECTOR AND QUANTUM CLASSIFIERS FOR USER 2 WITH SNR=10DB ON A QUANTUM SIMULATOR.

Classifiers	Encoding	Pauli	Train	Validation	Test
QKC	(11)	[Y, I, ZY]	93%	85%	52%
VQC RealAmp.	(11)	[X, ZY]	86%	80%	59%
VQC TwoLocal	(11)	[Z, I, ZY]	97%	85%	55%
Classical SVM	RBF	-	80%	65%	61%

TABLE II
COMPARISON PERFORMANCE OF SVM DETECTOR AND QUANTUM CLASSIFIERS FOR USER 2 WITH SNR=15DB ON A QUANTUM SIMULATOR.

Classifiers	Encoding	Pauli	Train	Validation	Test
QKC	(16)	[X, ZX]	97%	90%	80%
VQC RealAmp.	(15)	[Y, X, ZX]	90%	80%	82%
VQC TwoLocal	(17)	[Y, ZX]	90%	85%	81%
Classical SVM	RBF	-	97%	100%	71%

TABLE III
COMPARISON PERFORMANCE OF SVM DETECTOR AND QUANTUM CLASSIFIERS FOR USER 2 WITH SNR=20DB ON A QUANTUM SIMULATOR.

Classifiers	Encoding	Pauli	Train	Validation	Test
QKC	(17)	[Y,X, ZX]	100%	100%	80%
VQC RealAmp.	(15)	[Y, ZX]	96%	95%	80%
VQC TwoLocal	(15)	[Y,I, ZZ]	100%	100%	79%
Classical SVM	RBF	-	93%	100%	80%

in a set of 950 new points that were not previously used in the training and validation phase. For 15 dB of SNR, the classical SVM outperform all other methods in the validation dataset, but presents a poorer result in the test phase, which may indicate difficulty in generalizing the model. For 20 dB of SNR, QKC, VQC RealAmp and Classical SVM achieved the same results in the test phase.

On average, the accuracy over all three datasets in the testing phase is 70.7% on the QKC quantum simulator; 73.4% on VQC RealAmplitude; 71.7% on VQC TwoLocal and 70.7% on the classical SVM. So, our experimental evaluation showed the quantum approach outperform their classical counterpart on average by 1 to 2% in terms of accuracy in two of three different scenarios considered.

V. CONCLUSIONS

In this paper we evaluate the performance of classical kernel-based SVM, provided by sklearn.svm.SVC with RBF kernel function and default hyperparameter configurations, and compare with two quantum methods: QKC and VQC. We execute the quantum methods on Qiskit Aer simulator, a Python-based quantum simulator of IBM Qiskit. In order to find a quantum circuit that could yields the best performance for a given dataset, we use six different feature maps and 48 sequences of Pauli matrices into the definition of the unitary operator (10). Our experiments were performed on relatively small datasets with low-dimensional data. However, the results show that quantum computing can already be successfully used for practical applications in machine learning problems. An interesting question would be to apply this approach to a higher-dimensional data, such that a MU-DS-CDMA system with number of user greater than two. This would imply that, with a greater number of resources, possibly larger quantum circuits and with a greater number of qubits would be required, making simulation in classical systems quite difficult. However, this challenge will be mitigated as the effective number of qubits in quantum systems increases in the near future.

AKNOWLEDGEMENTS

This work was partially financed by the program “INOVA-CEFET/RJ – PIBITI/CNPq".

REFERENCES

[1] L. Dai, B. Wang, Z. Ding, Z. Wang, S. Chen, and L. Hanzo, *A survey of non-orthogonal multiple access for 5G*, IEEE Commun. Surveys Tuts., vol. 20, no. 3, pp. 2294-2323, 3rd Quart., 2018.

[2] M. Kulhandjian, H. Kulhandjian, C. D'Amours, H. Yanikomeroglu, D. A. Pados and G. Khachatrian, *Low-Complexity Decoder for Overloaded Uniquely Decodable Synchronous CDMA*, in IEEE Access, v. 10, 2022.

[3] M. Kulhandjian, H. Kulhandjian, C. D'Amours, and L. Hanzo, *Low density spreading codes for NOMA systems and a Gaussian separability based design*, IEEE Access, vol. 9, pp. 33963-33993, 2021.

[4] M. Kulhandjian and C. D'Amours, *Design of permutation-based sparse code multiple access system*, in Proc. IEEE 28th Annu. Int. Symp. Pers., Indoor, Mobile Radio Commun. (PIMRC), Montreal, QC, Canada, 2017.

[5] S. J. Nawaz, S. K. Sharma, S. Wyne, M. N. Patwary and M. Asaduzzaman, *Quantum Machine Learning for 6G Communication Networks: State-of-the-Art and Vision for the Future*, in IEEE Access, v. 7, 2019.

[6] P. Botsinis, D. Alanis, Z. Babar, S. X. Ng and L. Hanzo, *Joint Quantum-Assisted Channel Estimation and Data Detection*, in IEEE Access, v. 4, 2016.

[7] P. Rebentrost, M. Mohseni and S. Lloyd, *Quantum Support Vector Machine for Big Data Classification*, V. 14, Americ. Phys. Soc., 2014.

[8] L. Jung-Sik, L. Jae-Wan, H. Jae-Jeong, C. Kyung-Taek. *Multi-User Detection using Support Vector Machines*, The Journal of Korean Institute of Communications and Information Sciences, V. 34, 2009.

[9] C. Cortes and V. Vapnik. *Support-vector networks*. Mach Learn. v. 20, 273, 1995.

[10] M. Peter, and I. A. Faisal. *Mathematics for Machine Learning*. Cambridge University Press, 2020.

[11] S. Chen, A. K. Samingan, and L. Hanzo. *Support Vector Machine Multiuser Receiver for DS-CDMA Signals in Multipath Channels*. IEEE TRANSACTIONS ON NEURAL NETWORKS, v. 12, May, 2001.

[12] M. Nielsen, and I. Chuang. *Quantum Computation and Quantum Information*. Cambridge University Press, 2010.

[13] M. Schuld and N. Killoran. *Quantum Machine Learning in Feature Hilbert Spaces*. Phys. Rev. Lett., v. 122, Feb, 2019.

[14] V. Havlíček, A. D. Córcoles, K. Temme, A. W. Harrow, A. Kandala, J. M. Chow, J. M. Gambetta. *Supervised learning with quantum-enhanced feature spaces*. Nature, Mar, 2019.

[15] Y. Liu, S. Arunachalam, K. Temme. *A rigorous and robust quantum speed-up in supervised machine learning.* Nat. Phys., v.17, 2021.

[16] M. Schuld. *Supervised quantum machine learning models are kernel methods*. Quantum Physics, v.1, 1-21, 2021.

[17] H. Buhrman, R. Cleve, J. Watrous, and R. De Wolf. *Quantum fingerprinting*. Phys. Rev. Lett. v. 87, 2001.

[18] Y. Suzuki, H. Yano, Q. Gao, S. Uno, T. Tanaka, M. Akiyama and N. Yamamot. *Analysis and synthesis of feature map for kernel-based quantum classifier*. Quantum Machine Intelligence, v. 2, 2019.

[19] J. Preskill. *Quantum Computing in the NISQ era and beyond*. Quantum, v. 2, 2018.

[20] T. Hubregtsen, D. Wierichs, E. Gil-Fuster, H. S. Peter-Jan, P. K. Faehrmann, J. J. Meyer. *Training quantum embedding kernels on near-term quantum computers*. Phys. Rev. A, v. 106, Oct, 2022.

[21] Qiskit Community. *Qiskit Machine Learning overview*. https://qiskit.org. (Last accessed: 24.04.2023).