POLITECNICO DI TORINO
Repository ISTITUZIONALE

Evaluating the computational advantages of the Variational Quantum Circuit model in Financial Fraud Detection

Original
Evaluating the computational advantages of the Variational Quantum Circuit model in Financial Fraud Detection / Tudisco, A., Volpe, D., Ranieri, G., Curato, G., Ricossa, D., Graziano, M., Corbelletto, D.. - In: IEEE ACCESS. - ISSN 2169-3536. - 12:(2024), pp. 102918-102940. [10.1109/ACCESS.2024.3432312]

Availability:
This version is available at: 11583/2991116 since: 2024-08-02T06:59:15Z

Publisher:
IEEE

Published
DOI:10.1109/ACCESS.2024.3432312

Terms of use:

This article is made available under terms and conditions as specified in the corresponding bibliographic description in the repository

Publisher copyright

(Article begins on next page)

11 October 2026

IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 18 June 2024, accepted 18 July 2024, date of publication 22 July 2024, date of current version 2 August 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3432312

RESEARCH ARTICLE

Evaluating the Computational Advantages of the Variational Quantum Circuit Model in Financial Fraud Detection

ANTONIO TUDISCO[1], (Graduate Student Member, IEEE), DEBORAH VOLPE[1], (Graduate Student Member, IEEE), GIACOMO RANIERI[2], GIANBIAGIO CURATO[2], DAVIDE RICOSSA[2], MARIAGRAZIA GRAZIANO[3], AND DAVIDE CORBELLETTO[2]

[1]Department of Electronics and Telecommunications, Politecnico di Torino, 10129 Turin, Italy
[2]Intesa Sanpaolo, 10121 Turin, Italy
[3]Department of Applied Science and Technology, Politecnico di Torino, 10129 Turin, Italy

Corresponding author: Antonio Tudisco (antonio.tudisco@polito.it)

ABSTRACT Home banking and digital payments diffusion has greatly increased in recent years. As a result, fraud has also dramatically grown, resulting in the loss of billions of dollars worldwide every year. Therefore, banks and financial institutions are required to offer clients increasingly effective and sophisticated services for illegal transaction detection. Machine learning strategies are commonly employed for this crucial application. However, classical models are not satisfactory enough in highly unbalanced classification tasks like fraud detection. Quantum machine learning, working intrinsically in a higher dimensional computation space thanks to superposition and entanglement, can express more complex models than its classical counterpart and thus could provide a significant advantage in identifying potential illegal transactions. This work aims to analyze the potential of quantum classifiers in the fraud detection context, focusing on the Variational Quantum Circuit (VQC) model. The study has been led by exploiting a dataset based on real transactions provided by Intesa Sanpaolo of 500000 items with 15 features. Considering the limitations of contemporary Noisy Intermediate-Scale Quantum (NISQ) computers and quantum simulators, the dataset has been reduced in the number of transactions and features, exploiting Principal Component Analysis (PCA). The results obtained have been compared on equal terms with those of the most commonly employed classical methods, such as Logistic Regression, Random Forest, XGBoost, Support Vector Machine, and Neural Networks, obtaining a better classification quality in terms of recall. Even though this work is preliminary, the results are encouraging and prove the quantum models' potential in highly unbalanced classification tasks.

INDEX TERMS Quantum machine learning, fraud detection, quantum data encoding, variational quantum circuits, anomaly detection, anti-fraud engine, quantum computing, transaction classification, hybrid quantum-classical computing, angle encoding, quantum neural network.

I. INTRODUCTION

The diffusion of home banking and digital payments has grown significantly in recent years. Unfortunately, **financial fraud** increases dramatically with these, provoking **billions of dollars of annual loss worldwide** [1]. Consequently, to guarantee service quality to their clients, banks and financial institutions actively seek methods that are even more effective and sophisticated for **identifying illegal transactions**.

The associate editor coordinating the review of this manuscript and approving it for publication was Wei Huang.

© 2024 The Authors. This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 License.
For more information, see https://creativecommons.org/licenses/by-nc-nd/4.0/

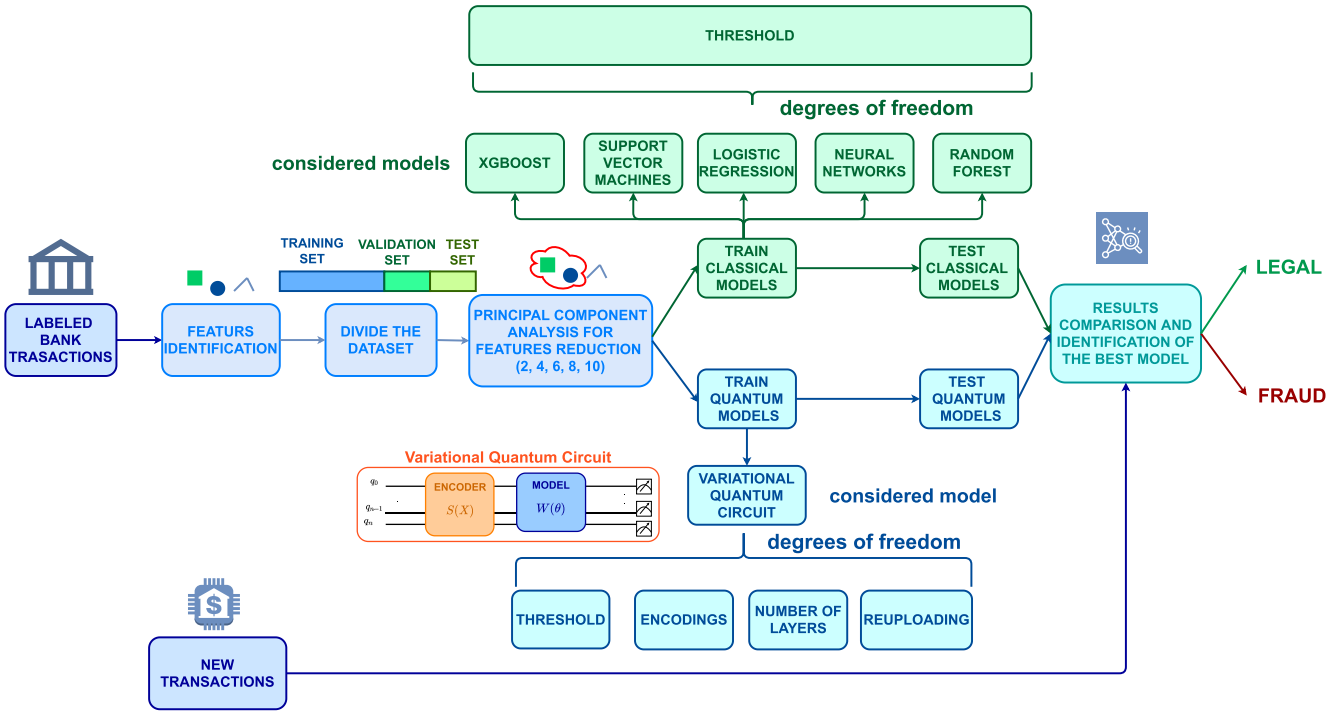


FIGURE 1. Starting from labeled financial transactions associated with fifteen features, obtained as alerts generated by the intesa sanpaolo transaction monitoring engine during 2022, the dataset has been split — exploiting a stratification technique for maintaining the proportion between the frauds and legal transactions — into training, validation, and test sets. Afterward, the Principal Component Analysis (PCA) technique was applied to reduce the number of features. Then, the training dataset has been exploited for training classical models, such as XGBoost, Support Vector Machine (SVM), logistic regression, neural networks, random forest, and quantum model, in particular, Variational Quantum Circuit (VQC). In both cases, the degrees of freedom, like the threshold, have been optimized to maximize the prediction quality in the validation dataset. The obtained results are compared for selecting the best model.

Machine learning (**ML**) is commonly employed for tackling this challenge [2], [3], [4], [5], recognizable as a binary classification task. However, classical models are not always wholly satisfactory when there is a **high imbalance between the two classes**. Specifically, the elements of one class are substantially more than those of the other. In recent decades, the attraction of quantum computing has reached an exceptional acceleration, especially in the realm of machine learning solutions [6], where its ability to perform a virtual parallel exploration of the solution spaces and handle complex models holds the promise to overcome classical limitations.

Indeed, **quantum machine learning** (**QML**) exploits **superposition** and **entanglement** for working inherently in a higher dimensional space. Moreover, the capability to express complex patterns can offer significant advantages in illegal transaction identification tasks. For these reasons, there has been a notable upsurge in interest in recent times regarding exploring QML for applications in this domain [7], [8].

This article delves into a comprehensive exploration of the **potential of quantum classifiers for fraud identification**, focusing on the **Variational Quantum Circuit** (**VQC**) model. The analysis leverages a **dataset based on real transactions** comprising about 500000 entries provided by **Intesa Sanpaolo Bank**, with **fifteen identified features**. To align with the constraints of contemporary **Noisy Intermediate-Scale Quantum** (**NISQ**) computers, which are affected significantly by the noise that compromises the obtained results, the quantum models have been executed on an ideal simulator. The complexity of simulations increases exponentially with the number of qubits, requiring the number of transactions and features reduction, using the classical **Principal Component Analysis** (**PCA**) procedure [9], for liming the complexity.

For each reduced dataset, 400 different models are trained and evaluated to identify the one that provides the best performance. Moreover, a comparative analysis has been performed, pitting the proposed quantum model against main classical methods, e.g., **Logistic Regression**, **XGBoost**, **Support Vector Machine**, **Random Forest**, and **Neural Networks**. On average, the quantum models exhibited a better classification quality in terms of precision and recall, emphasizing their potential efficacy in addressing highly unbalanced classification tasks, especially in datasets with a reduced number of features and elements.

Finally, the impact of noise in real quantum devices has been evaluated by performing predictions on `ibm_brisbane` real device.

The article is organized as follows. Section II presents the fundamentals of quantum computing and quantum machine

learning. In Section III, the proposed quantum model is presented and detailed. Section IV presents the implemented approach for optimizing the considered models with regard to the threshold hyperparameter. Section V shows the results and explains figures of merit and the validation methodology. Finally, in Section VI, conclusions are drawn, and future perspectives are illustrated.

II. THEORETICAL FOUNDATIONS

This section briefly presents the basis of the **Quantum Computing** (**QC**) paradigm (Section II-A) and its application in the **Machine Learning** (**ML**) context (Section II-B), named **Quantum Machine Learning** (**QML**), for developing classification models more complex than their classical counterpart. Further details about QC, ML, and QML can be found in [10], [11], [12], and [13], respectively.

Additionally, this section presents an analysis available in the literature about the exploitation of quantum computers for fraud detection classification tasks.

A. QUANTUM COMPUTING

Quantum computing is an innovative computational paradigm that leverages principles from quantum mechanics, such as superposition and entanglement, to accelerate some specific tasks.

The main differences between quantum and classical computing are:

- **probabilistic** nature of QC in contrast to classical computing that operates within a deterministic paradigm;
- quantum information **cannot be copied** (non-cloning theorem);
- the fundamental unit of quantum information, the **qubit**, can assume **infinite possible states** according to the superposition principle;
- a **quantum circuit** is a **time series of transformations** (gates) applied to the quantum system instead of physically constructed in a spatial layout as in classical circuits;
- all the quantum gates are **reversible**.

In the following, qubit and quantum gate concepts are introduced.

1) QUBIT

The fundamental unit of quantum information is the qubit. The generic state of a qubit $|\psi\rangle$ can be expressed, according to Dirac notation, as:

$$|\psi\rangle = c_0\,|0\rangle + c_1\,|1\rangle = c_0\begin{pmatrix}1\\0\end{pmatrix} + c_1\begin{pmatrix}0\\1\end{pmatrix} = \begin{pmatrix}c_0\\c_1\end{pmatrix}, \quad (1)$$

where $|0\rangle$ and $|1\rangle$ are the basis states 0 and 1, respectively, c_0 and c_1 are complex number called probability amplitudes.

It is possible to notice that, as previously mentioned, unlike classical computing, where a bit is constrained to assume 0 or 1, a qubit, ruled by the **superposition** principle, can exist in **any linear combination of its basis states**, offering infinite possible states. Regardless, observing a qubit (measure), it collapses into either of the two computational bases, $|0\rangle$ and $|1\rangle$, with a probability of $|c_0|^2$ and $|c_1|^2$, respectively. Consequently, the probability amplitudes must satisfy the following relation:

$$|c_0|^2 + |c_1|^2 = 1\,. \quad (2)$$

2) QUANTUM CIRCUITS

In general, the state vector $|\psi\rangle$ of N-qubit system can be obtained by extending the single qubit representation, through the tensor product of the state of the single qubits constituting the system:

$$\begin{aligned}|\psi\rangle &= |\psi_{N-1}\rangle \otimes |\psi_{N-2}\rangle \otimes \cdots \otimes |\psi_1\rangle \otimes |\psi_0\rangle \\ &= \begin{pmatrix}c_{0_{N-1}}\\c_{1_{N-1}}\end{pmatrix} \otimes \begin{pmatrix}c_{0_{N-2}}\\c_{1_{N-2}}\end{pmatrix} \otimes \cdots \otimes \begin{pmatrix}c_{0_0}\\c_{1_0}\end{pmatrix} \\ &= \begin{pmatrix}c_{00\cdots00}\\c_{00\cdots01}\\\vdots\\c_{11\cdots10}\\c_{11\cdots11}\end{pmatrix} \\ &= c_{00\cdots00}\,|00\cdots00\rangle + c_{00\cdots01}\,|00\cdots01\rangle \\ &\quad + \cdots + c_{11\cdots10}\,|11\cdots10\rangle + c_{11\cdots11}\,|11\cdots11\rangle\,, \end{aligned} \quad (3)$$

where the probability amplitude $c_{00\cdots00}$ is associated with the $|00\cdots00\rangle$, $c_{00\cdots01}$ to $|00\cdots01\rangle$ and so forth.

The N-qubit system state can be transformed by applying quantum gates, which can be formally described as unitary $2^n \times 2^n$ unitary matrices, where $n \leq N$. Gates involving at least two qubits (unitary matrices larger than 2×2) are exploited for creating **entanglement**, i.e. to establish a **strong correlation** between qubits where the state of one depends on the state of another.

Noteworthy quantum gates include:

- the **Pauli gates**, whose matrices are:

$$X = \begin{pmatrix}0&1\\1&0\end{pmatrix} Y = \begin{pmatrix}0&-i\\i&0\end{pmatrix} \quad Z = \begin{pmatrix}1&0\\0&-1\end{pmatrix}; \quad (4)$$

- the **rotational gates**, which are mostly used for QML applications, can be derived as exponential of the Pauli gates, and depend on the parametric angle θ:

$$\begin{aligned} R_x(\theta) &= e^{-i\theta X/2} = \begin{pmatrix}\cos(\frac{\theta}{2}) & -i\sin(\frac{\theta}{2})\\ -i\sin(\frac{\theta}{2}) & \cos(\frac{\theta}{2})\end{pmatrix} \\ R_y(\theta) &= e^{-i\theta Y/2} = \begin{pmatrix}\cos(\frac{\theta}{2}) & -\sin(\frac{\theta}{2})\\ \sin(\frac{\theta}{2}) & \cos(\frac{\theta}{2})\end{pmatrix} \\ R_z(\theta) &= e^{-i\theta Z/2} = \begin{pmatrix}e^{-i\frac{\theta}{2}} & 0\\ 0 & e^{i\frac{\theta}{2}}\end{pmatrix}; \end{aligned} \quad (5)$$

- **CNOT gate**, a controlled version of the X gate, applied to one qubit (the target) based on the state of another (the control). Its matrix is, if the control qubit is the Least Significant Qubit (LSQ) and the target is the Most

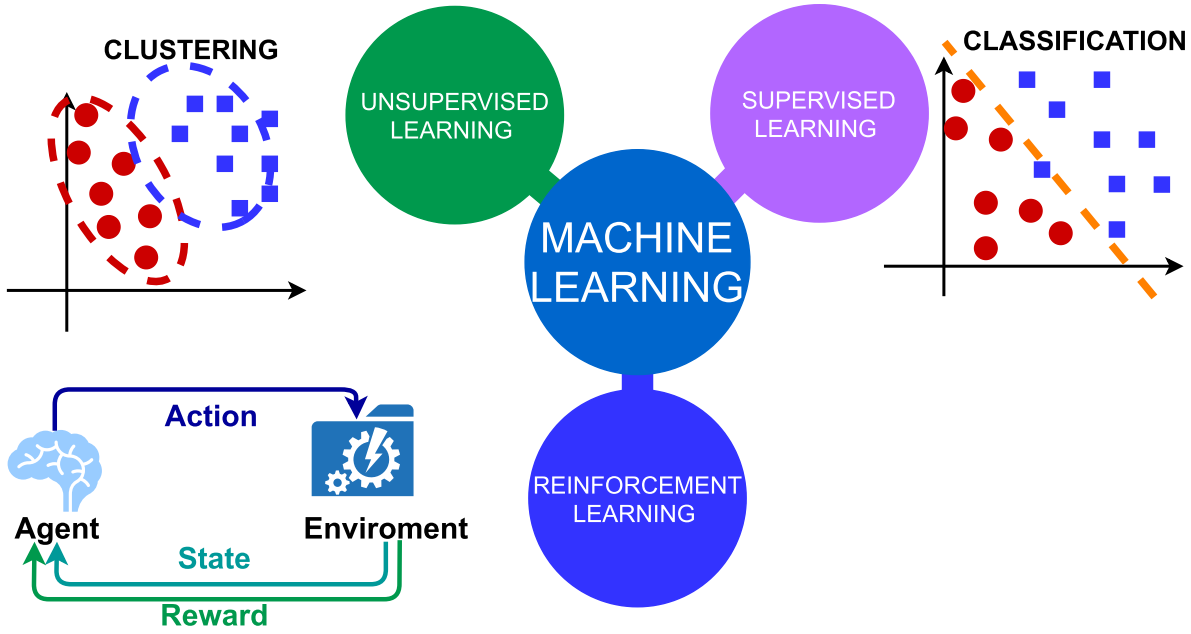


FIGURE 2. Representation of the different kinds of learning: supervised, unsupervised, and reinforcement. In the supervised block, a classification example is represented, while for the unsupervised it is shown as an example of clustering.

Significant Qubit (MSQ), the following:

$$\text{CNOT} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}. \tag{6}$$

B. MACHINE LEARNING

Machine Learning (**ML**) is a domain of **Artificial Intelligence** (**AI**), whose target is to develop models for adapting a computer system's behavior to the input for performing a specific task. In particular, the system becomes able to recognize patterns and make decisions or predictions based on data.

It can be subdivided, as shown in Figure 2, into three classes:

- **Supervised Learning**, in which the model is trained through a set of labeled data provided in input in the training phase (for example regression and classification);
- **Unsupervised Learning**, where the data provided as input in the training phase are not labeled (for example principal component analysis [14] and clustering [15]);
- **Reinforcement Learning**, in which the model is inserted in a feedback mechanism such that it can learn from both input data and experience.

Classification, which is the focus of this work, belongs to the Supervised Learning category. It aims to subdivide input data into two or more predefined classes (in the fraud detection context, fraud and legal transactions classes). There are three different types of classification: **binary**, **multi-class**, and **multi-label**. The difference between them is in the nature of output labels. In the first, the target is one of the two possible mutually exclusive classes; in the second, the possible classes are more than two and always mutually exclusive, while in the last, the possible classes can be non-mutually exclusive.

For performing a classification, it is necessary to define and train a model by evaluating a **loss function** and minimizing it through a proper optimizer, as shown in Figure 3. A model can be described as a black box that acquires the inputs and processes them by applying a parametric function to produce the desired outcome. The parameters of the function are chosen by defining the loss function as the distance between the predicted and the expected labels. This must be minimized with a proper optimizer to reduce errors.

The effectiveness of classification strongly depends on the model choice, which is, to all effects, a **hyper-parameter**, i.e. a parameter that is not derived from the learning process.

Quantum computation can be applied in this context to define new kinds of models, which can overcome the limitations of the existing classical ones in describing the target phenomena.

In this work, various classical machine learning models are considered, including:

- **Logistic Regression**;
- **Random Forest**;
- **eXtreme Gradient Boosting**;
- **Support Vector Machine**;
- **Neural Network**.

Each of these models will be systematically addressed in the subsequent sections.

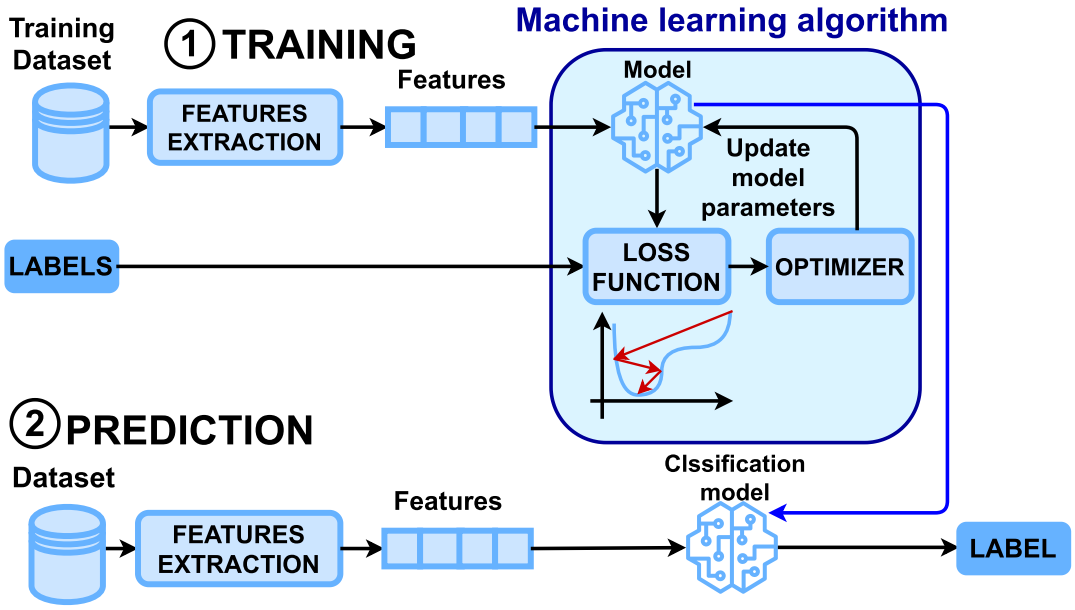


FIGURE 3. Classification block scheme.

1) LOGISTIC REGRESSION

Logistic Regression [16], a statistical technique, is employed for binary classification tasks. It stands as a foundational model within the realm of machine learning. Given a set of independent variables, the logistic regression model endeavors to estimate the probability that a specific instance pertains to a designated class. It is characterized by the logistic function, defined as follows:

$$f(x, w) = \frac{1}{1 + e^{-w \cdot x}}, \tag{7}$$

where $f(x, w)$ denotes the probability of observing a positive outcome, and w represents the coefficients to be determined. This function ensures that the predicted probabilities reside within the closed interval [0, 1].

2) RANDOM FOREST

Random Forest [17] is an ensemble classifier comprised of multiple decision trees, collectively determining the class based on the aggregated results of individual trees. For each of the K trees in the forest, a subset of features and training data points are selected.

This modeling approach exhibits the potential to achieve high accuracy and scalability. However, Random Forest models may be susceptible to **overfitting**, i.e., they learn too much from training data, capturing noise instead of real patterns, leading to poor performance on new, unseen data.

3) EXTREME GRADIENT BOOSTING

XGBoost [18], an acronym of **eXtreme Gradient Boosting**, stands as a formidable and versatile machine learning algorithm within the ensemble learning domain. It exhibits notable efficacy in tasks involving regression and classification.

At its essence, XGBoost operates as an ensemble of decision trees, with each tree dedicated to correcting the errors of its predecessor. The algorithm utilizes a gradient-boosting framework, iteratively enhancing performance by optimizing a predefined loss function by adding decision trees to the ensemble.

The predictive model is represented as a summation of individual tree predictions:

$$\hat{y}_i = \sum_{k=1}^{K} f_k(x_i), \tag{8}$$

where, $\hat{y}_i$ signifies the predicted output, K denotes the number of trees, and $f_k(x_i)$ represents the prediction of the k-th tree for input x_i.

XGBoost is distinguished by its incorporation of regularization techniques, effective handling of missing values, and the integration of features. Notably, it demonstrates computational efficiency and has garnered popularity in several ML competitions.

4) NEURAL NETWORKS

A **Neural Network** [19] is a computational model inspired by the structure and functioning of the human brain. It comprises interconnected nodes, often organized into **layers**. The fundamental unit is the **artificial neuron**, which receives input, processes it using a set of weights, and produces an output.

Mathematically, the output of a neuron is determined by an **activation function** applied to the weighted sum of its inputs:

$$y = f\left(\sum_{i=1}^{n} w_i \cdot x_i + b\right), \tag{9}$$

where w_i represents the weights, x_i denotes the inputs, b the bias term, and f the activation function.

Neural Networks typically consist of an **input layer**, one or **more hidden layers**, and an output layer. Learning in a Neural Network involves adjusting the weights and biases based on the error between the predicted output and the actual target.

This model is characterized by high accuracy and robustness to noise and outliers, although it generally requires an extended training period.

5) SUPPORT VECTOR MACHINE

A **Support Vector Machine** (**SVM**) [20] represents a powerful machine learning model designed primarily for binary classification tasks. The main goal of an SVM is to discern patterns within data by identifying a hyperplane that effectively separates two distinct classes, **maximizing the margin between the classes and ensuring optimal classification performance**.

In situations where a linear decision boundary proves insufficient to capture the underlying complexity of the data, SVMs offer the flexibility to operate in higher-dimensional spaces by employing kernel functions, such as the **Radial Basis Function** (RBF) kernel. These kernel functions facilitate the transformation of the input data into a space where a linear separation is more achievable, allowing SVMs to handle nonlinear relationships effectively.

C. PREVIOUS WORKS

In recent years, some explorations of quantum machine learning solutions for fraud detection tasks have been presented, like [9], [21], and [22], aiming to evaluate the potential of quantum computers in real-world applications. It is important to note that the mentioned works have been published in the last three years, proving that nowadays fraud detection is an important topic.

For example, in [9], quantum support vector machines and variational quantum algorithms classification capabilities have been compared against the Logistic Regression, Decision Tree, Random Forest, K-Nearest Neighbors, and SVM classical models. Despite employing a synthetic dataset, the study demonstrated a quantum advantage. Similar to our work, this has also reduced the dimensionality of the dataset through appropriate techniques.

In [21], Quantum Graph Neural Networks have been explored considering an available credit card fraud detection dataset and compared with Classical Graph Neural Networks, demonstrating the benefit of the quantum models. Notably, the consideration of thresholds as model hyperparameters aligns with our methodology.

On the other hand, [22] explores the potential benefit of quantum computers in machine learning model optimization. In particular, it exploits quantum annealers, a special-purpose quantum computer for combinatorial optimization.

The main strengths of our work compared to the current literature are:

- employment of a real dataset provided by Intesa Sanpaolo bank consisting of recent (two years ago) transactions;
- comprehensive analysis of the dataset characteristics;
- optimization of the encoding mechanism of Variational Quantum Circuit, considered as a model hyperparameter;
- optimization of the model threshold as a hyperparameter, enhancing the model capability of managing imbalance;
- comparisons with the main classical models;
- evaluation of execution on a real quantum device and ideal behavior obtained with a simulator.

III. PROPOSED QUANTUM MODELS

The model explored in this work is called **Variational Quantum Circuit** (**VQC**) [12], [23]. The key concept behind VQC is the refinement of parameters guided by an objective function. It is a hybrid quantum-classical approach composed of a parameterized quantum circuit (quantum part), representing the model, and an optimizer (classical part) exploited for a loss-function-dependent evaluation of the parameters (Figure 4).

The quantum circuit consists of two parts. The first one, named **encoder**, properly describes the **data features** by embedding them into a **quantum state**. At the same time, the second is a **parameterized** quantum circuit, that modifies the initial state vector allowing the classification of the input data. The final quantum states and classification results are evaluated through the **measurement** operation.

In the training phase, the **classical optimizer** aims to optimize the quantum circuit parameters according to the values measured encoding input data in the quantum circuit and the chosen loss function. Once the model has been fully trained for classifying new data, it is sufficient to evaluate the measurement obtained by embedding it in the quantum circuit with the trained parameters.

Each part of the deployed models is detailed in the following.

A. ENCODING CIRCUIT

The **encoder** plays the fundamental role of embedding classical data into quantum states. This operation is performed through a unitary matrix $U(x)$. In the literature, several types of encoding strategies have been proposed and discussed. The main are **angle**, **amplitude**, and **basis encoding**. The first encodes the data inside the **parametric angle of rotational gates**, requiring N qubit for N features. At the same time, the second embeds the data, after a **normalization step**, into the **amplitudes of a quantum state**, requiring $\log_2(N)$ qubits. Despite the reduced number of qubits required, implementing this strategy requires several transformations, which increase exponentially with the qubit count, overcoming those required for angle encoding. Moreover, the depth is also

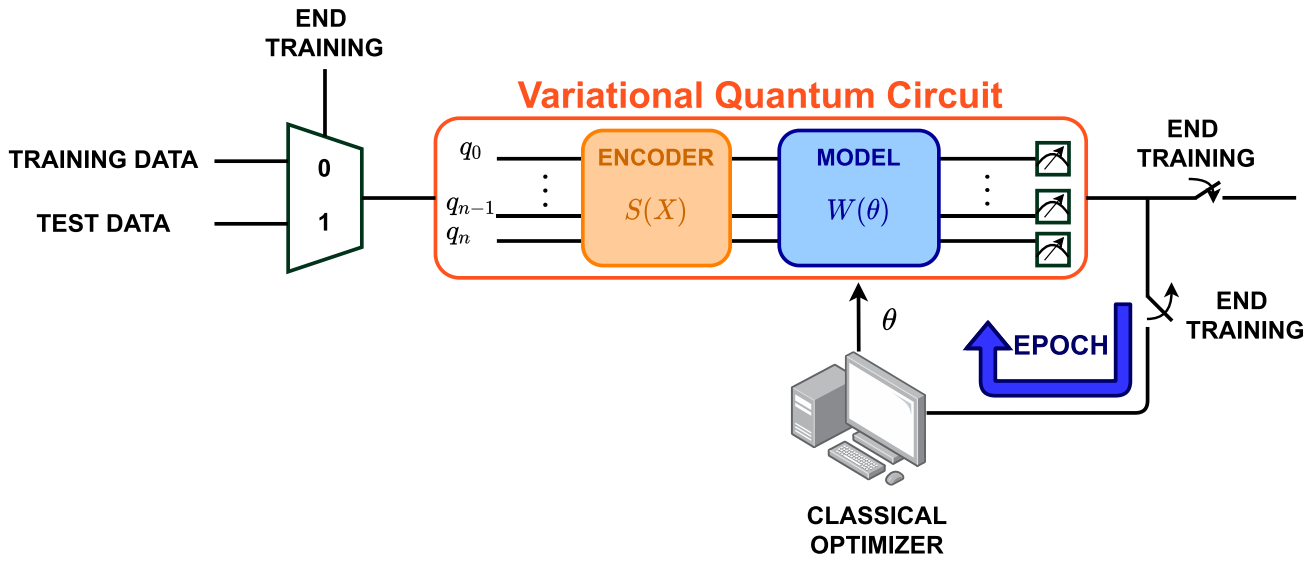


FIGURE 4. Variational Quantum Circuit (VQC) model, composed of a parametric quantum circuit and a classical optimizer.

significantly higher than one of the angle methods, affecting the fidelity significantly in the case of execution on real hardware. Finally, the last associates each feature, written as a binary string, with a **basis state**, requiring $N \times M$ qubits, where M is the number of bits in the binary string.

This work explores **angle encoding**, which can be considered the most promising, since the qubit scaling is acceptable for near-term applications, considering the limitations of the current NISQ devices, and requires a lower number of transformations than the amplitude one [24].

To better understand how angle encoding works, an example is reported in the following.

1) EXAMPLE

Considering a vector of features $x = [x_1\ x_2\ x_3\ x_4]$ and the RY gates for encoding, the quantum state obtained is equal to:

$$|\psi\rangle = \otimes_{i=1}^{n} |\psi_i\rangle\ , \tag{10}$$

where

$$|\psi_i\rangle = \cos\left(\frac{x_i}{2}\right)|0\rangle + \sin\left(\frac{x_i}{2}\right)|1\rangle\ . \tag{11}$$

The graphical representation of this quantum circuit is shown in Figure 5.

Exploiting the RY gate is only one of many possibilities for implementing angle encoding. Indeed, all the rotational gates RX, RY, and RZ and their combinations can be employed, applying or not a uniform superposition layer before. As proven in [24], not all the possible combinations are meaningful. The combinations considered in this work are reported in Table 1.

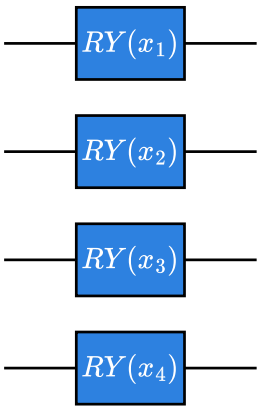


FIGURE 5. Angle encoding of a four-feature vector exploiting RY gates.

TABLE 1. Angle encoding strategies which are grouped by the number of transformations required.

1-gate	2-gates	3-gates	4-gates
RX	RX-RY	RX-RY-RZ	H-RY-RX-RZ
RY	RX-RZ	RX-RZ-RY	H-RY-RZ-RX
	RY-RX	RY-RX-RZ	H-RZ-RX-RY
	RY-RZ	RY-RZ-RX	H-RZ-RY-RX
	H-RY	H-RY-RX	
	H-RZ	H-RY-RZ	
		H-RZ-RX	
		H-RZ-RY	

B. PARAMETRIZED CIRCUIT

The parametrized circuit is the core of the VQC model since it processes the encoded data. It usually consists of two-qubit gates to create a correlation among features and rotational gates, whose angles represent the model parameters.

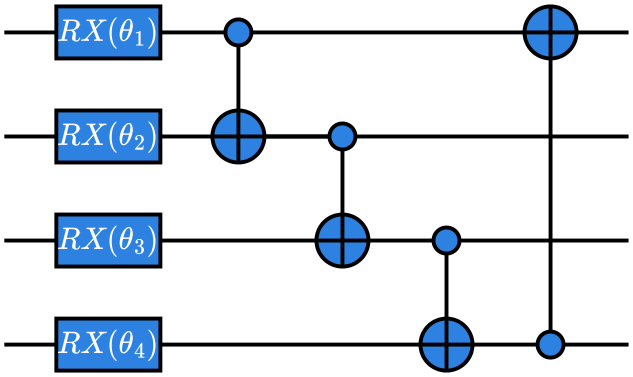


FIGURE 6. Single layer of a four-qubit basic entangling layer circuit. The number of parameters is equal to the number of qubits.

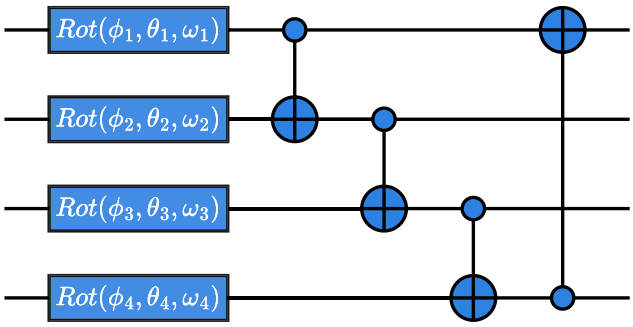


FIGURE 7. Single layer of a four-qubit strongly entangling layer circuit. the number of parameters is equal to three times the number of qubits.

Among all possible types of ansatz, this study is focused on **Strongly Entangling Layer** and the **Basic Entangler Layer** circuits, both available in the PennyLane library [25].

The Basic Entangling Layer circuit, shown in Figure 6, consists of one-parameter (θ_i) single-qubit rotations (usually RX gates) on each qubit, followed by a so-called *ring* of CNOT gates. On the other hand, the Strongly Entangling Layer, as shown in Figure 7, includes three-parameter ($\phi_i, \theta_i, \omega_i$) single-qubit rotations (ROT gate) on each qubit — equivalent to RZ-RY-RZ — and the CNOT ring for creating an entanglement layer. Thanks to its three angles, the ROT gate, which is a generic unitary operation, guarantees an higher flexibility in the transformation, permitting the rotation in any direction of the Bloch sphere. In this way, a higher expressivity of the model should be ensured. The drawback is that the number of parameters to optimize increases, potentially making convergence more complex. Therefore, the best option is problem-dependent, making the parametrized circuit a further hyperparameter of the model. For this reason, both circuits have been evaluated in the presented analysis.

C. RE-UPLOADING

The **re-uploading** [26] is a strategy employed for training VQC. It consists of repeatedly applying the data encoding layers and parametric blocks, as shown in Figure 8. It is

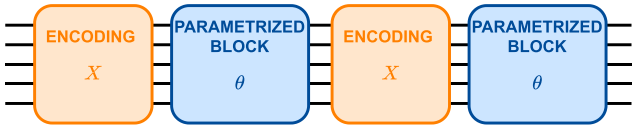


FIGURE 8. Re-uploading technique.

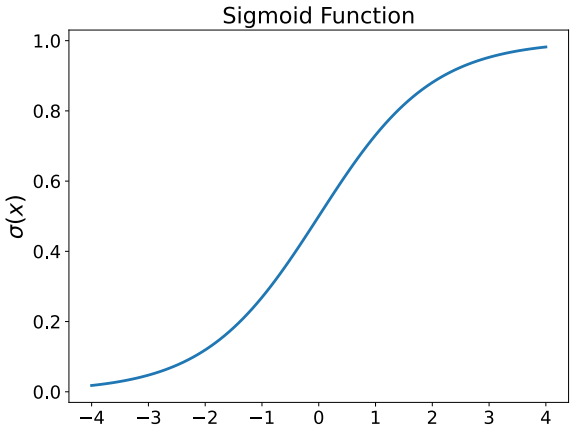


FIGURE 9. Sigmoid function $\sigma(x) = \frac{1}{1+e^{-x}}$.

commonly used to enhance the model's expressive power and its ability to represent complex data patterns, leading to higher accuracy and more robust performance. However, its main disadvantages include longer computational times due to its iterative nature and increased sensitivity to noise compared to simpler versions of the quantum algorithm caused by the increased depth. As a result, deciding whether to apply it or not becomes an additional model hyperparameter, necessitating evaluation of the model under both conditions to achieve the best possible performance.

D. MEASUREMENT

In order to obtain the classification result, it is necessary to **measure** the most significant qubit. The exploited observable is **Pauli-Z**, whose eigenvalues are 1 and -1, associated with the eigenvectors $|0\rangle$ and $|1\rangle$, respectively. Therefore, the output obtained is in the range $[-1, 1]$ that has to be added to a trainable bias b and then normalized in the interval $[0, 1]$, through the sigmoid function, shown in Figure 9, whose expression is:

$$\sigma(x) = \frac{1}{1 + e^{-x}}. \tag{12}$$

where x is the output of the model combined with the bias parameter. Therefore, the value is normalized in the interval [0,1].

The choice of the Pauli-Z operator for measurement is related to its straightforward implementation, direct interpretation, and null overhead in terms of quantum operations, which is more efficient in the case of execution on real hardware.

E. CLASSICAL OPTIMIZER

After defining the quantum circuit, the model parameters must be updated to enhance classification performance. For parameter optimization, gradient descent optimizers are commonly used, necessitating the computation of the derivative of the loss function for each parameter. Consequently, back-propagation techniques [27], which are available in libraries such as Pennylane and Pytorch, are employed to evaluate these derivatives. However, these techniques are only applicable when the quantum circuit is simulated. Indeed, alternative methods, such as parameter shift rule, are required in case of execution on real hardware for derivative computation.

IV. THRESHOLD OPTIMIZATION

In the case of unbalanced classification, optimizing the threshold could be fundamental for accurately classifying the elements belonging to the minority class, specifically class 1. Therefore, a proper criterion should be defined. In the literature, several figures of merit exist with advantages and disadvantages, which have been discussed in the following.

A. FIGURES OF MERIT

First, **accuracy**, the most renowned Figure of Merit, is defined as the ratio between correctly classified samples and the overall dataset dimension:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}, \tag{13}$$

where **TP** represents the count of **true positives**, **TN** signifies the count of **true negatives**, **FP** denotes the count of **false positives**, and **FN** reflects the count of **false negatives**. Unfortunately, this is inappropriate for highly unbalanced datasets as achieving a high score can be obtained by predicting all the elements belonging to the majority class.

Another crucial Figure of Merit is called **recall** or sensitivity. It expresses the effectiveness of predicting the positive class and can be computed as follows:

$$\text{recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}. \tag{14}$$

It can be employed for imbalance classification tasks where accurately classifying positives is of essential importance, and the penalties introduced by false positives are considered negligible, implementing a **conservative policy**.

Its counterpart for negative samples is known as **specificity** or selectivity and can be calculated as follows:

$$\text{specificity} = \frac{\text{TN}}{\text{TN} + \text{FP}}. \tag{15}$$

Starting from recall and specificity, a figure of merit that balances both can be defined:

$$\text{Gmean} = \sqrt{\text{recall} \cdot \text{specificity}}, \tag{16}$$

This could be a good compromise, but it gives the same importance to both, which could be risky in high-imbalance classification.

Another commonly employed figure of merit is **precision**, defined as the ratio between the count of true positives and the total number of positive instances:

$$\text{precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}, \tag{17}$$

and it expresses the **effectiveness of the model in identifying positive samples**.

Precision and recall can be combined to derive the following composite figures of merit:

$$\text{F1Score} = \frac{2 \cdot \text{recall} \cdot \text{precision}}{\text{recall} + \text{precision}}, \tag{18}$$

$$\text{F}\beta\text{Score} = \frac{(1 + \beta^2) \cdot \text{recall} \cdot \text{precision}}{\text{recall} + \beta^2 \cdot \text{precision}}, \tag{19}$$

where β can be employed for balancing their contributions as functions of the task characteristics.

Alternatively, another interesting figure of merit for imbalance dataset classification is the **positive likelihood ratio**, which can be defined as:

$$\text{LR+} = \frac{\frac{\text{TP}}{\text{TP+FN}}}{\frac{\text{FP}}{\text{FP+TN}}}. \tag{20}$$

Beyond the threshold figures of merit, ranking metrics can be exploited for evaluating a model's capability of separating classes of the dataset of interest. These provide a comprehensive perspective by showing the performance of two figures of merit across various threshold values.

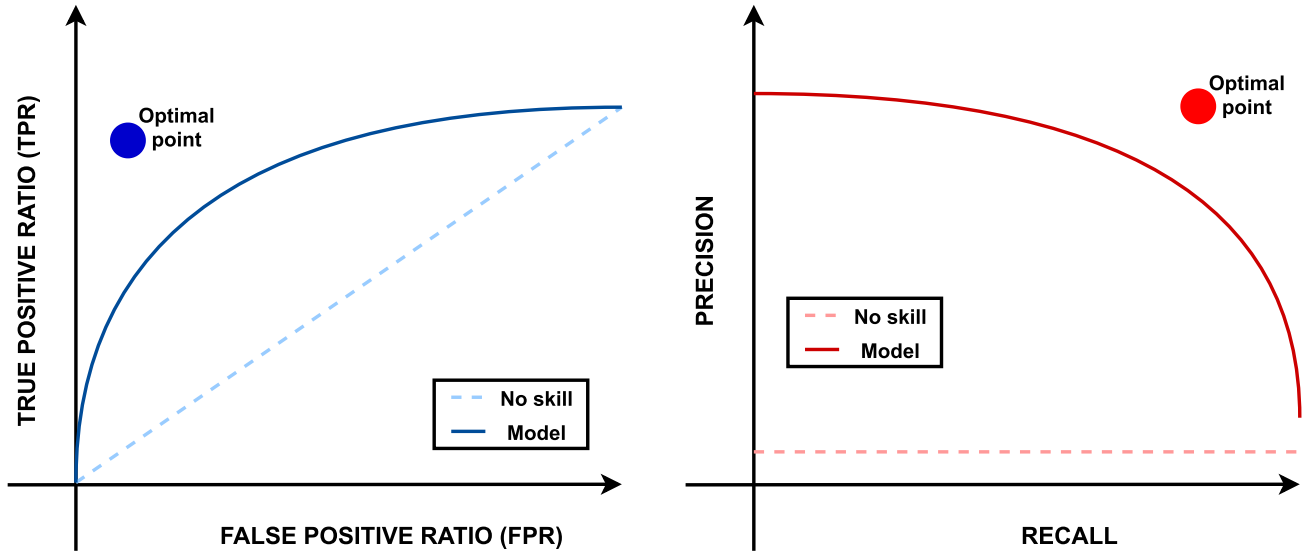
The most popular figure of merit in this context is the **Receiver Operating Characteristic** (ROC) curve, which shows the **True Positive Ratio** (TPR), i.e., the recall, and the **False Positive Ratio** (FPR) under **different thresholds**. An example of its expected shape is provided in Figure 10a. However, it is proven that this curve cannot be particularly meaningful in the case of the highly unbalanced dataset since it provides a too-optimistic view of the model performance, giving the same importance to both classes.

Alternatively, the **precision-recall** curve can be exploited. It can be exploited similarly, but it **focuses on problem performance in classifying the minor class** (the fraudulent transaction in this case). Indeed, this plot shows the precision and the recall across diverse thresholds. Figure 10b gives an example of the expected results for an effective model.

In both cases, the area under the curves can be calculated to obtain a single score, summarizing the plot that can be used for comparing different models.

B. METHODOLOGY

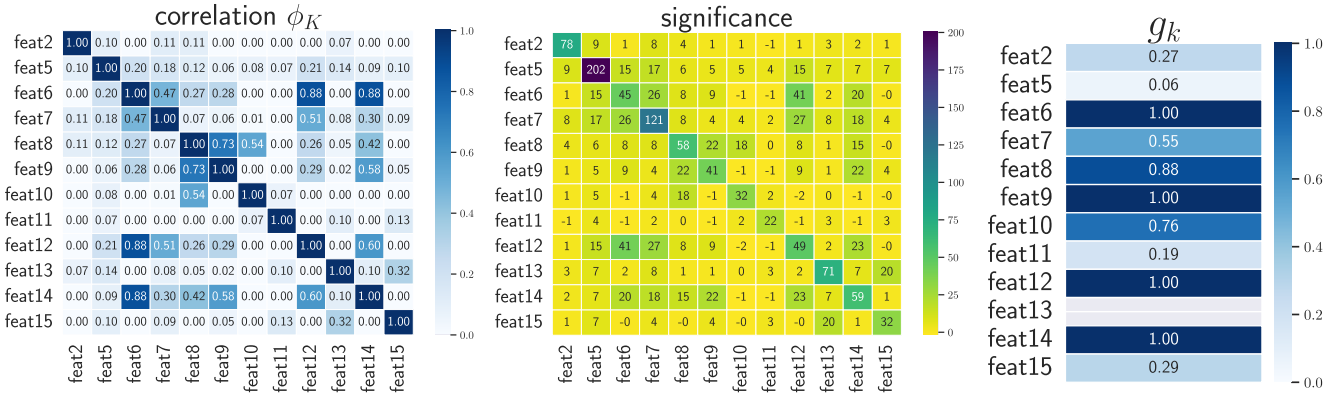
This work evaluates all the presented figures of merit for each considered classical or quantum model for **fifty threshold values** in the range (0, 0.5] in the training and validation set. The samples in the range do not have a uniform distribution. Indeed, most of the samples are concentrated in the range (0, 0.05] since, considering the imbalance in the dataset, this region is the most meaningful.



(a) Receiver Operating Characteristic (ROC) curve, which aims to summarize the performance of a binary classifier model across different threshold values. This graphical figure of merit plots the True Positive Rate (TPR), i.e. the recall, against the False Positive Rate (FPR) for each threshold setting. The dotted line represents the expected behaviour for a non-skilled classifier, contrasting with the blue line, which characterises an effective model. The optimal point to attain is in the top-left corner, while the points in the bottom-right are the least desirable.

(b) Precision-recall curve, whose target is to provide a summary of a binary classifier model effectiveness with different values of threshold. This graphical representation shows the precision against the recall for each considered threshold value. The horizontal dotted line, assuming a value equal to the ratio of positive samples, is the expected result in the case of a non-skilled classifier. In contrast, the red line shows the behaviour of an effective classifier, with the ideal classifier depicted as a red dot in the top-right corner. At the same time, results below the horizontal dotted line are the worst possible outcomes.

FIGURE 10. **Explanation through examples of the ranking metrics for comparing models.**



(a) ϕ_K correlation matrix defined in [28]

(b) Significance overview

(c) Global features correlation

FIGURE 11. **Complete dataset analysis.**

The area under the ROC curve and precision-recall curves have been exploited as key metrics for evaluating the efficacy of quantum models generated through various encoding mechanisms alongside classical models in training and validation steps. These metrics are recognized as comprehensive indicators of model performance.

Given the application's critical nature and the dataset's inherent class imbalance, we prioritize the precision-recall curve over the ROC curve. Consequently, the optimal encoding technique and the number of layers have been identified for each dataset size — both in terms of features and data — based on its ability to **maximize precision-recall score on the validation dataset**. Subsequently, the threshold maximizing the Fβ-score with β equal to two in the validation set has been found for the best quantum and classical models. The Fβ-score

has been chosen since it considers both precision and recall.

Finally, the optimized models have been evaluated with the test dataset, considering precision and recall figures of merit. In this way, quantum and classical models can be compared to identify the best model for the application.

V. RESULTS

This section presents the analysis of the employed datasets and the outcomes obtained by exploiting classical and quantum models. Additional results, not detailed within this manuscript, are available through the GitHub repository.

A. SETTINGS

The employed models have been defined exploiting the Pennylane library (version 0.28), an open software framework for differentiable programming for quantum computers, which is particularly useful for Quantum Machine Learning and Quantum Chemistry applications owing to its compatibility with classical machine learning libraries such as Tensorflow and Pytorch. Notably, in this work, the models are trained with Pytorch (version 0.28).

In terms of training methodology, all developed models adhere to the same approach. This approach involves training each model for ten epochs utilizing a mini-batch strategy with batches of size 10. The Adam optimizer [29], a stochastic gradient descent optimizer, has been selected for this purpose, with an initial learning rate set to 0.01.

The evaluated loss function to be minimized is the **Binary Cross Entropy**, defined as:

$$L(p, y) = \frac{1}{M} \sum_{i=0}^{M-1} (-y_n \log p_n - (1 - y_n) \log(1 - p_n)), \tag{21}$$

where y are the target labels, p are the outputs of the VQC, passed through the sigmoid function, and M is the batch size.

The quantum models have been trained by exploiting the **default.qubit** Pennylane simulator running on the Legion HPC server accessible through HPC Polito services. Prediction results on training, test and validation datasets have been performed on single-process Intel(R) Xeon(R) Gold 6134 CPU @ 3.20 GHz opta-core, Model 85, with a memory of about 103 GB [30].

The classical models have been trained on a single-process Intel(R) Xeon(R) Gold 6134 CPU @ 3.20 GHz opta-core, Model 85, with a memory of about 103 GB [30]. Prediction outcomes on training, test and validation datasets have been performed on the same platform.

B. DATASET

The labelled dataset employed for the presented analysis comprises **515651** entries associated with **fifteen features**. These data samples represent **alerts generated by the Intesa Sanpaolo Transaction Monitoring Engine during 2022**, where **sensible information has been hashed with the SHA-512** algorithm. As previously mentioned, the dataset is strongly unbalanced since the ratio between the samples belonging to the 1 class, which identifies the frauds, and the overall data set is equal to 2%.

Each element of the dataset consists of the following numerical features:

- transaction timestamp;
- transaction amount;
- year;
- month;
- day.

In addition, there are eleven categorical features, each hashed for privacy considerations.

Undoubtedly, hashing some categories leads to a loss of information. However, the dataset used closely mirrors real-world scenarios as it is not a purely synthetic dataset but rather derived from actual transactions of Intesa Sanpaolo, with specific information intentionally protected for privacy reasons.

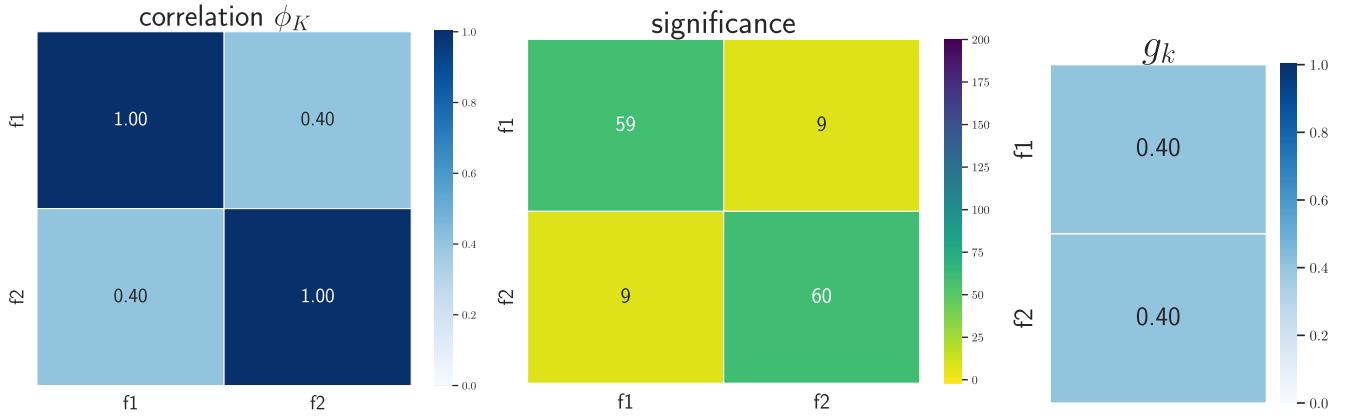
The overall provided dataset has been analyzed to evaluate the ϕ_K correlation among features — able also to show non-linear relations and to work with both categorical and interval features — and the significance of these correlations, through the PhiK library [28]. The conducted analysis is summarized in Figure 11, where the correlation matrix, the global correlation of each feature, and the significance of feature correlations are reported. The feature associated with years was excluded from this analysis since all the data comes from transactions that occurred in the year 2022, rendering this element non-contributory to the examination.

Current quantum computers, characteristic of the Noisy Intermediate-Scale Quantum (NISQ) era, present strong limits in terms of reliability. Furthermore, the inherent non-idealities present in quantum technologies, such as relaxation and decoherence, can significantly compromise the achieved results, especially in circuits with high depth or involving numerous qubits, not allowing the evaluation of the QML potential.

Exploring quantum solutions through simulators is today the most common strategy. In this way, the possibility of obtaining non-meaningful results due to noise is avoided. However, the time required for simulation grows exponentially with the number of qubits. Unfortunately, the dataset dimension and the number of features are incompatible with the quantum simulation, due to its exponential increase of complexity.

Consequently, some dataset reductions are necessary to evaluate the prospect of quantum computing in this classification task.

Therefore, the **Principal Component Analysis** (**PCA**) technique can be applied, after a proper normalization step, mandatory for its effectiveness. The PCA is an unsupervised learning approach able to reduce the dimensionality of datasets while preserving crucial information. In particular, it computes the covariance matrix of the dataset, extracts its eigenvectors and the eigenvalues, filters the eigenvectors to



(a) ϕ_K correlation matrix defined in [28] (b) Significance overview (c) Global features correlation

FIGURE 12. Reduced two-feature 1000-transaction dataset.

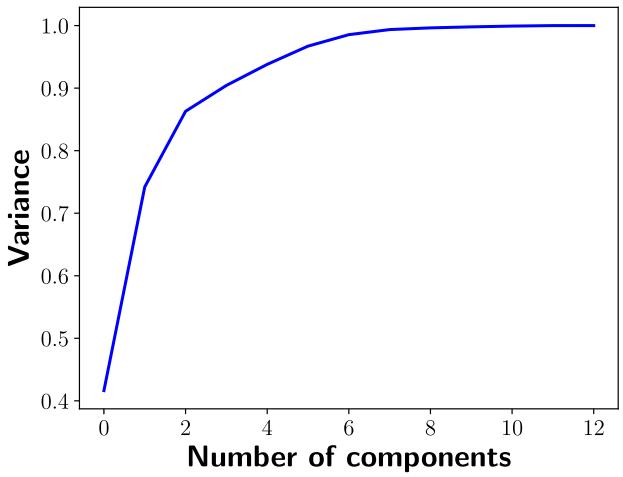


FIGURE 13. The cumulative explained variance for each component, applying principal component analysis. Notice that eight features include more than 99% of the variance.

match the desired number of components, sorting them by their associated eigenvalue, and multiply the original space by the generated feature vector. A drawback of this approach is the loss of interpretability of the dataset, which can be a significant limitation for some applications.

In order to identify reasonable numbers of components to keep with PCA, the cumulative explained variance can be evaluated and plotted, as shown in Figure13. It is possible to notice that eight components include more than 99% of the variance, i.e. about all the meaningful information in the dataset. For this reason, we have decided not to consider datasets with more than eight components. In particular, we consider two, four, six, and eight components for the reduced datasets. Moreover, the dataset dimension has been reduced through a proper sampling technique, which maintains the ratio among elements of the two classes, resulting in dataset sizes of 1000, 2000, 4000, 6000, 8000, 10000, and 20000 transactions.

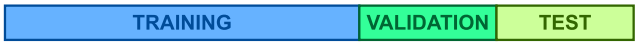


FIGURE 14. Dataset split in training, validation and test set.

The reduced datasets have been analyzed deeply with the same technique applied to the original dataset. For resuming the obtained results, the correlations and significances of the smallest (two features and 1000 transactions) and the larger (eight features and 20000 transactions) analyzed datasets are reported in Figures 12 and 15, respectively. It is possible to notice that the reduced datasets have a high level of Φ_k correlations, with high significance scores.

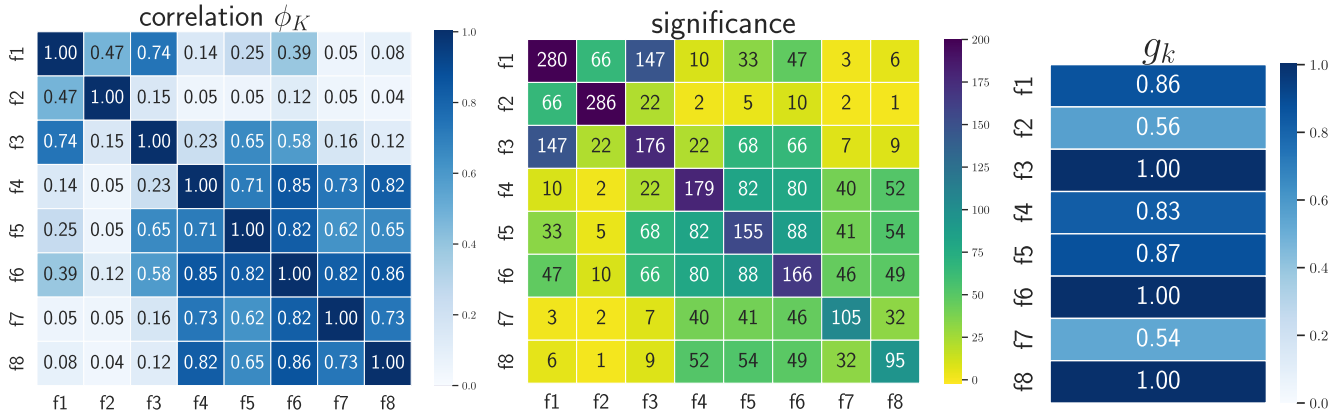
As usual, each obtained reduced dataset has been split — exploiting a stratification technique for maintaining the proportion between the frauds and legal transactions — into training, validation, and test sets equal to 80%, 10%, and 10%, respectively, of the overall dataset (Figure 14). The first has been employed for training the model, the second for selecting the optimal one and the last for evaluating the classification performance.

In order to acquire information as much information as possible about data, the two feature-reduced datasets are graphically represented. The smallest (1000 transactions) and largest (20000 transactions) are reported in Figure 16. It is possible to notice that the elements belonging to class 1 (frauds) and class 0 (legal transactions) are not linearly separable, which makes the classification more complex.

C. CLASSICAL MODELS RESULTS

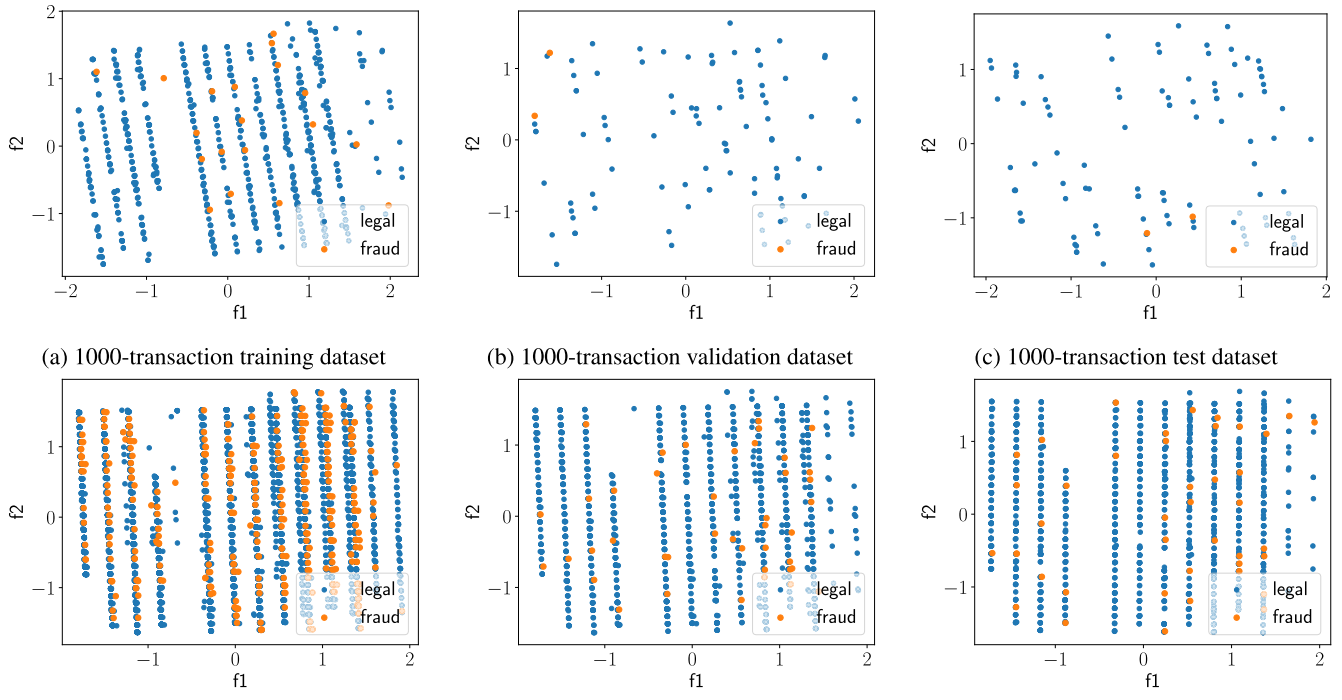
In this section, the results obtained with classical models in training and validation using various thresholds are discussed in terms of the area under the ROC curve and precision-recall (ROC PR) curves.

Table 2 displays the area under the ROC curve and precision-recall (PR) curves achieved with the **Logistic Regression model**. It is possible to notice that the ROC scores



(a) ϕ_K correlation matrix defined in [28] (b) Significance overview (c) Global features correlation

FIGURE 15. **Reduced eight-feature 20000-transaction dataset.**



(a) 1000-transaction training dataset (b) 1000-transaction validation dataset (c) 1000-transaction test dataset

(d) 20000-transaction training dataset (e) 20000-transaction validation dataset (f) 20000-transaction test dataset

FIGURE 16. **Graphical representation of two-feature datasets.**

obtained are, on average, quite high. However, precision-recall are very low, especially when the datasets with fewer features are considered. This means that this model is expected to provide, chosen the best threshold, high accuracy scores, but limited precision and recall, which are more crucial in this kind of application.

In Table 3, the area under the ROC curve and precision-recall (PR) curves obtained with different thresholds on the training and validation datasets using the **Random Forest model** are presented. Notably, the best outcomes are attained with a dataset containing eight and six features. However, it is observed that, generally, while the model exhibits a high ROC score during training, there is a substantial decrease in performance during validation, indicative of potential overfitting. Both in training and validation, this model presents quite low scores of precision-recall, indicating poor performance from the precision and recall point of view.

Moving to Table 4, the area under the ROC curve and precision-recall (PR) curves resulting from various thresholds on both training and validation datasets utilizing

TABLE 2. Area under the ROC curve and precision-recall (PR) curves obtained with the training and validation datasets, considering the logistic regression model. The best absolute results for each column are in blue, while the best results for each dataset size are bolded.

Dim	Feat	Training		Validation	
		ROC	PR	ROC	PR
1000	2	0.638	0.030	0.357	0.014
1000	4	0.833	0.080	0.671	0.027
1000	6	**0.999**	0.336	**1.000**	0.000
1000	8	0.998	**0.531**	**1.000**	**0.500**
2000	2	0.556	0.024	0.332	0.016
2000	4	0.656	0.035	0.358	0.015
2000	6	**0.997**	0.560	0.986	0.179
2000	8	**0.997**	**0.691**	**1.000**	**0.750**
4000	2	0.552	0.023	0.569	0.021
4000	4	0.997	0.298	0.997	0.199
4000	6	0.976	0.388	0.898	0.247
4000	8	**0.999**	**0.398**	**0.999**	**0.449**
6000	2	0.535	0.024	0.692	0.043
6000	4	0.826	0.153	0.899	0.156
6000	6	0.992	0.410	0.994	**0.463**
6000	8	**0.999**	**0.428**	**1.000**	0.333
8000	2	0.564	0.026	0.509	0.025
8000	4	0.661	0.051	0.718	0.062
8000	6	0.991	**0.452**	0.992	**0.496**
8000	8	**0.999**	0.351	**0.999**	0.321
10000	2	0.561	0.025	0.491	0.023
10000	4	0.814	0.113	0.797	0.105
10000	6	0.993	**0.446**	0.996	**0.435**
10000	8	**0.999**	0.324	**0.997**	0.413
20000	2	0.570	0.026	0.552	0.023
20000	4	0.734	0.077	0.727	0.096
20000	6	0.992	**0.432**	0.990	**0.454**
20000	8	**0.999**	0.193	**0.999**	0.130

TABLE 3. Area under the ROC curve and precision-recall (PR) curves obtained with the training and validation datasets, considering the random forest model. The best absolute results for each column are in blue, while the best results for each dataset size are bolded.

Dim	Feat	Training		Validation	
		ROC	PR	ROC	PR
1000	2	**1.000**	0.000	0.684	0.099
1000	4	**1.000**	0.000	0.944	0.309
1000	6	**1.000**	0.000	**1.000**	0.000
1000	8	**1.000**	0.000	**1.000**	**0.500**
2000	2	**1.000**	0.000	0.695	0.091
2000	4	**1.000**	0.000	0.599	0.028
2000	6	**1.000**	0.000	0.985	0.134
2000	8	**1.000**	0.000	**1.000**	**0.250**
4000	2	**1.000**	0.000	0.593	0.031
4000	4	**1.000**	0.000	0.990	0.081
4000	6	**1.000**	0.000	0.992	**0.208**
4000	8	**1.000**	0.000	**1.000**	0.000
6000	2	**1.000**	0.000	0.819	0.189
6000	4	**1.000**	0.000	0.948	**0.241**
6000	6	**1.000**	0.000	0.994	0.172
6000	8	**1.000**	0.000	**1.000**	0.000
8000	2	**1.000**	0.000	0.796	0.243
8000	4	**1.000**	**0.007**	0.955	0.222
8000	6	**1.000**	0.000	0.991	**0.254**
8000	8	**1.000**	0.000	**1.000**	0.059
10000	2	**1.000**	0.000	0.721	0.144
10000	4	**1.000**	0.000	0.994	**0.227**
10000	6	**1.000**	0.000	0.996	0.073
10000	8	**1.000**	0.000	**0.998**	0.000
20000	2	**1.000**	0.000	0.890	0.273
20000	4	**1.000**	0.000	0.995	**0.250**
20000	6	**1.000**	0.000	0.998	0.090
20000	8	**1.000**	0.000	**0.999**	0.023

the **XGBoost** model are shown. The best performance, in terms of precision-recall in validation, with this model is observed with a dataset consisting of 8000 transactions and eight features. Also in this case, while the ROC scores obtained are, on average, quite high, precision-recall results are very low, especially in the datasets with fewer features.

In Table 5, the area under the ROC curve and precision-recall (PR) curves obtained with different thresholds on both training and validation datasets using the **support vector machine model**, considering both **linear** and **radial basis function** (rbf) kernels, are presented. For the linear kernel, the distance is evaluated as the inner product of the two feature vectors:

$$k(x, x') = x \cdot x', \tag{22}$$

while for the radial basis function (RBF) kernel, the distance is evaluated as:

$$k(x, x') = e^{-\frac{\|x-x'\|^2}{2\sigma^2}}. \tag{23}$$

The most favorable outcome, in terms of precision-recall in validation, in this case, is achieved with the dataset comprising 8000 transactions and features features. It is possible to notice that, on average, the rbf kernel performs perceptibly better than the linear one with datasets considering a lower number of features.

Furthermore, Table 6 illustrates the area under the ROC curve and precision-recall (PR) curves resulting from various thresholds on both training and validation datasets utilizing the **Neural Network** model. The model with the number of layers maximizing the area under the precision-recall curve in validation is reported for each dataset. Remarkably, the optimal performance in terms of precision-recall in validation is observed with a dataset containing 20000 transactions and four features.

Upon comparing all the classical results, it is observable that **logistic regression** emerges as the most promising model for addressing the task at hand, while XGBoost is the worst from the area below precision-recall in the validation point of view. In all the classical models considered, it is possible to notice that the worst performance has been obtained with the smallest datasets. This is coherent with the expectation since these contain less information for optimizing the model.

D. QUANTUM MODELS RESULTS

In this section, the results obtained with quantum models in training and validation using various thresholds are discussed

TABLE 4. Area under the ROC curve and precision-recall (PR) curves obtained with the training and validation datasets, considering the XGBoost model. The best absolute results for each column are in blue, while the best results for each dataset size are bolded.

Dim	Feat	Training		Validation	
		ROC	PR	ROC	PR
1000	2	**1.000**	0.000	0.250	0.012
1000	4	**1.000**	0.000	0.696	**0.266**
1000	6	**1.000**	0.000	**1.000**	0.000
1000	8	**1.000**	0.000	**1.000**	0.000
2000	2	**1.000**	**0.053**	0.543	**0.077**
2000	4	**1.000**	0.026	0.628	0.029
2000	6	**1.000**	0.000	0.985	0.000
2000	8	**1.000**	0.000	**1.000**	0.000
4000	2	0.998	**0.148**	0.456	0.016
4000	4	**1.000**	0.000	0.996	0.082
4000	6	**1.000**	0.013	0.980	**0.149**
4000	8	**1.000**	0.000	**0.997**	0.000
6000	2	0.983	**0.242**	0.676	0.062
6000	4	0.999	0.105	0.994	0.156
6000	6	0.999	0.021	0.994	**0.173**
6000	8	**1.000**	0.000	**1.000**	0.000
8000	2	0.956	**0.213**	0.668	0.095
8000	4	0.996	0.189	0.934	**0.293**
8000	6	**1.000**	0.013	0.993	0.107
8000	8	**1.000**	0.000	**1.000**	0.176
10000	2	0.910	**0.234**	0.506	0.018
10000	4	0.998	0.136	0.964	**0.112**
10000	6	0.999	0.009	0.995	0.105
10000	8	**1.000**	0.000	**0.998**	0.043
20000	2	0.828	0.212	0.657	0.070
20000	4	0.991	**0.223**	0.994	**0.211**
20000	6	0.999	0.034	0.996	0.138
20000	8	**1.000**	0.000	**0.999**	0.000

in terms of the area under the ROC and precision-recall (PR) curves. To streamline the discussion, only the model with the encoding mechanism and the number of layers maximizing the area under the precision-recall curve in validation for each dataset are reported. Those models are considered the best for this task, in particular, since they should guarantee good results in terms of precision and recall in the test set. In the case of parity, the least complex model, i.e., with the lowest number of layers and gates involved in the encoding mechanism, has been considered. The other results obtained with the other encoding mechanism and the number of layers are available in the GitHub repository.

Table 7 shows the area under the ROC curve and precision-recall (PR) curves resulting from various thresholds with the **basic entangling layer variational quantum circuit**. The ROC scores obtained in training are, on average, lower than those obtained by classical models, but the precision-recall scores, especially in validation, are promising and significantly better. Moreover, it could be highlighted that the encoding mechanisms guaranteeing the best performance are the most complex ones, i.e., including more than one rotational gate or at least the Hadamard layers.

Table 8 shows the area under the ROC curve and precision-recall (ROC PR) curves obtained by employing the **strongly entangling layer variational quantum circuit**. Also in this case, while the outcomes obtained in training are not significantly higher, the precision-recall scores in validation

TABLE 5. Area under the ROC curve and precision-recall (PR) curves obtained with the training and validation datasets, considering the support vector machine model with linear and radial basis function (rbf) kernel functions. The best absolute results for each column are in blue, while the best results for each dataset size are bolded.

Dim	Feat	Kernel	Training		Validation	
			ROC	PR	ROC	PR
1000	2	linear	0.373	0.015	0.730	0.031
1000	2	rbf	0.701	0.060	0.411	0.015
1000	4	linear	0.577	0.027	0.513	0.016
1000	4	rbf	0.991	**0.771**	0.957	0.344
1000	6	linear	0.998	0.047	**1.000**	0.000
1000	6	rbf	0.998	0.047	**1.000**	**0.500**
1000	8	linear	0.999	0.050	**1.000**	**0.500**
1000	8	rbf	**0.999**	0.050	**1.000**	0.000
2000	2	linear	0.500	0.011	0.500	0.010
2000	2	rbf	0.695	0.036	0.338	0.014
2000	4	linear	0.363	0.015	0.598	0.022
2000	4	rbf	0.557	0.415	0.515	0.010
2000	6	linear	0.996	**0.321**	0.987	**0.237**
2000	6	rbf	0.996	0.276	0.984	0.204
2000	8	linear	**0.999**	0.024	**1.000**	0.000
2000	8	rbf	**0.999**	0.000	**1.000**	0.250
4000	2	linear	0.500	0.011	0.500	0.010
4000	2	rbf	0.529	0.023	0.497	0.014
4000	4	linear	0.995	0.147	0.994	0.085
4000	4	rbf	0.996	0.147	0.994	0.164
4000	6	linear	0.983	0.393	0.979	0.292
4000	6	rbf	0.980	**0.359**	0.981	**0.317**
4000	8	linear	**0.999**	0.024	**1.000**	0.125
4000	8	rbf	**0.999**	0.063	0.997	0.000
6000	2	linear	0.531	0.024	0.685	0.041
6000	2	rbf	0.533	0.023	0.572	0.024
6000	4	linear	0.693	0.099	0.785	0.061
6000	4	rbf	0.948	0.354	0.896	**0.351**
6000	6	linear	0.985	**0.377**	0.992	0.348
6000	6	rbf	0.992	0.347	0.992	0.311
6000	8	linear	**0.999**	0.041	**1.000**	0.000
6000	8	rbf	**0.999**	0.041	**1.000**	0.000
8000	2	linear	0.444	0.017	0.516	0.019
8000	2	rbf	0.584	0.027	0.514	0.021
8000	4	linear	0.514	0.021	0.558	0.025
8000	4	rbf	0.939	**0.541**	0.928	**0.509**
8000	6	linear	0.991	0.436	0.991	0.445
8000	6	rbf	0.991	0.366	0.993	0.366
8000	8	linear	**0.999**	0.061	**1.000**	0.176
8000	8	rbf	**0.999**	0.074	**1.000**	0.114
10000	2	linear	0.512	0.021	0.440	0.017
10000	2	rbf	0.501	0.019	0.474	0.018
10000	4	linear	0.516	0.029	0.642	0.040
10000	4	rbf	0.876	**0.417**	0.867	**0.392**
10000	6	linear	0.994	0.406	0.996	0.365
10000	6	rbf	0.995	0.373	0.994	0.335
10000	8	linear	**0.999**	0.030	**0.998**	0.043
10000	8	rbf	**0.999**	0.045	0.997	0.118
20000	2	linear	0.553	0.027	0.507	0.020
20000	2	rbf	0.541	0.023	0.433	0.016
20000	4	linear	0.462	0.018	0.467	0.018
20000	4	rbf	0.820	0.316	0.765	0.342
20000	6	linear	0.992	**0.417**	0.988	**0.430**
20000	6	rbf	0.992	0.313	0.989	0.381
20000	8	linear	**0.999**	0.030	0.999	0.046
20000	8	rbf	**0.999**	0.030	**1.000**	0.066

TABLE 6. Area under the ROC curve and precision-recall curves (PR) obtained with the training and validation datasets, considering the neural network model with the number of layers maximizing the ROC PR in validation. The best absolute results for each column are in blue, while the best results for each dataset size are bolded.

Dim	Feat	Layer	Training		Validation	
			ROC	PR	ROC	PR
1000	2	6	0.657	0.033	0.291	0.013
1000	4	7	0.849	**0.098**	0.712	**0.031**
1000	6	10	0.998	0.000	1.000	0.000
1000	8	11	**1.000**	0.000	**1.000**	0.000
2000	2	5	0.606	0.036	0.477	0.019
2000	4	8	0.680	0.050	0.415	0.016
2000	6	8	0.996	**0.129**	0.987	**0.250**
2000	8	9	**1.000**	0.025	**1.000**	**0.250**
4000	2	5	0.570	0.028	0.614	0.025
4000	4	8	0.989	0.144	0.992	0.085
4000	6	10	0.989	**0.319**	0.981	**0.272**
4000	8	3	**0.998**	0.034	**1.000**	0.125
6000	2	4	0.546	0.022	0.706	0.037
6000	4	8	0.931	**0.284**	0.975	**0.275**
6000	6	7	**0.993**	0.255	0.993	0.147
6000	8	6	0.959	0.138	**1.000**	0.083
8000	2	5	0.559	0.025	0.501	0.026
8000	4	7	0.897	**0.311**	0.838	**0.272**
8000	6	8	0.990	0.243	0.990	0.235
8000	8	11	**0.998**	0.030	**1.000**	0.059
10000	2	2	0.563	0.025	0.481	0.029
10000	4	6	0.942	**0.550**	0.916	**0.477**
10000	6	10	0.993	0.209	0.994	0.168
10000	8	3	**0.997**	0.084	**0.997**	0.158
20000	2	2	0.568	0.025	0.546	0.023
20000	4	4	0.829	**0.322**	0.818	**0.542**
20000	6	7	0.990	0.220	0.991	0.263
20000	8	8	**0.999**	0.084	**1.000**	0.115

TABLE 7. Area under the ROC curve and precision-recall curves (PR) obtained with the training and validation datasets, considering the basic entangling layer variational quantum circuit with the encoding mechanism and number of layers maximizing the ROC PR in validation. The best absolute results for each column are in blue, while the best results for each dataset size are bolded.

Dim	Feat	Layer	Enc	Reup	Training		Validation	
					ROC	PR	ROC	PR
1000	2	2	Z_H	True	**0.552**	0.024	0.793	0.268
1000	4	10	ZY_H	True	0.424	0.018	0.543	0.071
1000	6	10	ZYX_H	False	0.542	**0.026**	**1.000**	**0.750**
1000	8	10	ZYX_H	False	0.451	0.019	**1.000**	**0.750**
2000	2	2	YZ	False	0.491	0.019	0.656	0.087
2000	4	8	YX	True	0.477	0.020	0.493	0.055
2000	6	2	YZX_H	False	0.496	0.023	0.996	0.699
2000	8	6	ZXY_H	False	**0.565**	**0.029**	**1.000**	**0.875**
4000	2	10	YX_H	False	**0.558**	**0.023**	0.677	0.064
4000	4	10	XYZ	False	0.455	0.020	**0.999**	**0.907**
4000	6	8	ZY_H	True	0.503	0.020	0.975	0.422
4000	8	6	Z_H	True	0.510	0.020	0.993	0.845
6000	2	10	XZY	True	0.550	0.026	0.635	0.201
6000	4	6	XZ	False	0.449	0.019	0.958	0.407
6000	6	6	Y_H	True	0.454	0.018	0.993	0.825
6000	8	4	YZ_H	False	**0.560**	**0.028**	**1.000**	**0.958**
8000	2	8	YXZ	True	0.500	**0.021**	0.560	0.039
8000	4	6	ZXY_H	False	0.497	**0.021**	0.848	0.129
8000	6	6	ZY_H	False	0.477	0.019	0.993	0.740
8000	8	6	Y_H	False	**0.507**	**0.021**	**0.998**	**0.905**
10000	2	8	YXZ	False	**0.518**	**0.021**	0.536	0.043
10000	4	2	YXZ	False	0.492	**0.021**	0.824	0.549
10000	6	2	XYZ	True	0.460	0.019	0.992	0.753
10000	8	4	YX_H	True	0.496	0.020	**0.994**	**0.857**
20000	2	8	ZY_H	True	**0.507**	0.021	0.560	0.038
20000	4	6	ZYX_H	False	0.506	**0.029**	0.737	0.436
20000	6	10	Y_H	False	0.499	0.021	0.916	0.595
20000	8	6	YXZ	False	0.493	0.020	**0.983**	**0.927**

outperform all the classical models, proving the ability of quantum computers to model more complex scenarios better.

As in the article [24], it is clear that re-uploading can be considered a hyper-parameter in the definition of the model. In fact, considering the different reduced datasets, not always the re-uploading models outperform others.

E. COMPARISON OF CLASSICAL AND QUANTUM MODELS IN TEST

Afterward, the analysis of the models in training and validation set, the threshold maximizing the **Fβscore** ($\beta = 2$), which takes into account both precision and recall, in validation has been identified for each model for each dataset. Considering these thresholds, the prediction outcomes obtained by considering the **test dataset** have been obtained. Precision and recall figures have been evaluated and collected in Table 9 for comparisons. We would like to remark that, in this particular application, these are the most relevant Figures of Merits since they show the capability of **correctly identifying the positive outcomes** (recall), i.e. the frauds, **without exceeding in costly false positive allarms** (precision).

The classical models present poor performance, especially in terms of precision, for datasets with a few number of features (two and four), while quantum models guarantee better results. On average, it is possible to notice that **quantum models provide good results in terms of precision and recall** with respect to classical models, **representing in the majority of the cases the best compromise among the two metrics**. This proves the potential of QML in overcoming classical models in terms of prediction quality in the case of unbalanced classification tasks.

To be precise, the best compromise between precision and recall is:

- for **1000 elements** and **two features** dataset is the **VQC model with basic entangling layer considering Z_H encoding and 2 layers with re-uploading technique**, guaranteeing the best precision and a recall close to the best;
- for **1000 elements** and **four features** dataset is the **VQC model with strongly entangling layer considering ZY_H encoding and 10 layers with re-uploading technique**, guaranteeing the best precision and a recall close to the best;

TABLE 8. Area under the ROC curve and precision-recall curves (PR) obtained with the training and validation datasets, considering the strongly entangling layer variational quantum circuit with the encoding mechanism and number of layers maximizing the ROC PR in validation.

Dim	Feat	Layer	Enc	Reup	Training		Validation	
					ROC	PR	ROC	PR
1000	2	2	Z_H	True	0.552	0.024	0.793	0.268
1000	4	10	ZY_H	True	0.424	0.018	0.543	0.071
1000	6	10	ZYX_H	False	**0.542**	**0.026**	**1.000**	**0.750**
1000	8	10	ZYX_H	False	0.451	0.019	**1.000**	**0.750**
2000	2	2	YZ	False	0.491	0.019	0.656	0.087
2000	4	8	YX	True	0.477	0.020	0.493	0.055
2000	6	2	YZX_H	False	0.496	0.023	0.996	0.699
2000	8	6	ZXY_H	False	**0.565**	**0.029**	**1.000**	**0.875**
4000	2	10	YX_H	False	**0.558**	**0.023**	0.677	0.064
4000	4	10	XYZ	False	0.455	0.020	**0.999**	**0.907**
4000	6	8	ZY_H	True	0.503	0.020	0.975	0.422
4000	8	6	Z_H	True	0.510	0.020	0.993	0.845
6000	2	10	XZY	True	0.550	0.026	0.635	0.201
6000	4	6	XZ	False	0.449	0.019	0.958	0.407
6000	6	6	Y_H	True	0.454	0.018	0.993	0.825
6000	8	4	YZ_H	False	**0.560**	**0.028**	**1.000**	**0.958**
8000	2	8	YXZ	True	0.500	**0.021**	0.560	0.039
8000	4	6	ZXY_H	False	0.497	**0.021**	0.848	0.129
8000	6	6	ZY_H	False	0.477	0.019	0.993	0.740
8000	8	6	Y_H	False	**0.507**	**0.021**	**0.998**	**0.905**
10000	2	8	YXZ	False	**0.518**	**0.021**	0.536	0.043
10000	4	2	YXZ	False	0.492	**0.021**	0.824	0.549
10000	6	2	XYZ	True	0.460	0.019	0.992	0.753
10000	8	4	YX_H	True	0.496	0.020	**0.994**	**0.857**
20000	2	2	YZX	False	0.498	0.021	0.575	0.038
20000	4	4	ZYX_H	False	0.513	0.021	0.770	0.615
20000	6	4	XY	True	**0.517**	**0.022**	0.978	0.639
20000	8	2	ZX_H	True	0.481	0.020	**0.998**	**0.908**

- for **1000 elements** and **six features** dataset, **logistic regression** model provides the best precision and the best recall;
- for **1000 elements** and **eight features** dataset is the **VQC model with basic entangling layer considering ZYX_H encoding and 10 layers without re-uploading** technique, guaranteeing the best precision and a recall close to the best;
- for **2000 elements** and **two features** dataset is the **VQC model with basic entangling layer considering YZ encoding and 2 layers without re-uploading technique**, guaranteeing the best precision and recall;
- for **2000 elements** and **four features** dataset is the **VQC model with strongly entangling layer considering YX encoding and 2 layers with re-uploading** technique, assuring the best precision and recall;
- for **2000 elements** and **six features** dataset is the **VQC model with basic entangling layer considering YZX_H encoding and 2 layers without re-uploading** technique, providing the best precision and a recall close to the best;
- for **2000 elements** and **eight features** dataset **logistic regression, random forest and support vector machine with linear kernel** give the best precision and recall;
- for **4000 elements** and **two features** dataset is the **VQC model with basic entangling layer considering YX_H encoding and 10 layers without re-uploading** technique, providing the best precision and a recall close to the best;
- for **4000 elements** and **four features** dataset is the **VQC model with basic entangling layer considering XYZ encoding and 10 layers without re-uploading** technique, providing the best precision and recall;
- for **4000 elements** and **six features** dataset is the **VQC model with basic entangling layer considering ZY_H encoding and 8 layers with re-uploading** technique, providing the best precision and recall;
- for **4000 elements** and **eight features** dataset **support vector machine with linear kernel** gives the best precision and recall;
- for **6000 elements** and **two features** dataset is the **VQC model with strongly entangling layer considering XZY encoding and 10 layers with re-uploading** technique, providing the best precision and recall;
- for **6000 elements** and **four features** dataset is the **VQC model with basic entangling layer considering XZ encoding and 6 layers without re-uploading** technique, providing the best precision and recall;
- for **6000 elements** and **six features** dataset is the **VQC model with strongly entangling layer considering YZ_H encoding and 4 layers without re-uploading** technique, guaranteeing the best precision and recall;
- for **6000 elements** and **eight features** dataset is the **VQC model with basic entangling layer considering YZ_H encoding and 4 layers without re-uploading** technique, providing the best precision and a recall close to the best;
- for **8000 elements** and **two features** dataset is the **VQC model with basic entangling layer considering YXZ encoding and 8 layers with re-uploading** technique, providing the best precision and a recall close to the best;
- for **8000 elements** and **four features** dataset is the **VQC model with strongly entangling layer considering ZXY_H encoding and 6 layers without re-uploading** technique, providing the best precision and recall;
- for **8000 elements** and **six features** dataset is the **VQC model with basic entangling layer considering ZY_H encoding and 6 layers without re-uploading** technique, providing the best precision and a recall close to the best;
- for **8000 elements** and **eight features** dataset is the **VQC model with basic entangling layer considering Y_H encoding and 8 layers with re-uploading** technique, providing a precision and a recall close to the best;
- for **10000 elements** and **two features** dataset is the **VQC model with strongly entangling layer considering YXZ encoding and 8 layers without re-uploading** technique, providing the best precision and recall;

TABLE 9. Precision (P) and Recall (R) obtained with the test datasets, considering Logistic Regression (LR), Neural Network (NN) with the optimal number of layers, Random Forest (RF), SVM with the linear kernel, SVM with the rbf kernel, XGBoost, Variational Quantum Circuit (VQC) with Basic entangling layer (maximizing the area under the precision-recall curves in the test dataset) and Variational Quantum Circuit (VQC) with strongly entangling layer (maximizing the area under the precision-recall curves in the test dataset). For each of them, the threshold maximizes the FβScore, with β set equal to 2. The best absolute results for precision and recall are in blue, while the suboptimal, allowing to reach a good compromise, are bolded.

Dim	Feat	LR		NN		RF		SVM linear		SVM rbf		XGBoost		VQC Basic		VQC Strongly	
		P	R	P	R	P	R	P	R	P	R	P	R	P	R	P	R
1000	2	0.000	0.000	0.000	0.000	0.000	0.000	0.034	**1.000**	0.022	**1.000**	0.000	0.000	**1.000**	**0.980**	0.959	0.979
1000	4	0.000	0.000	0.000	0.000	0.000	0.000	0.044	**1.000**	0.000	0.000	0.000	0.000	0.980	0.980	**1.000**	**0.980**
1000	6	**1.000**	**1.000**	**1.000**	0.500	**1.000**	0.500	0.400	1.000	0.667	**1.000**	0.000	0.000	**0.990**	**0.980**	0.541	0.964
1000	8	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	**1.000**	**0.980**	**0.969**	**0.990**
2000	2	0.000	0.000	0.000	0.000	0.062	0.250	0.020	1.000	0.019	0.750	0.000	0.000	**1.000**	**0.980**	0.985	0.980
2000	4	0.000	0.000	0.000	0.000	0.056	0.250	0.014	0.500	0.010	0.500	0.000	0.000	0.964	0.979	**0.995**	**0.980**
2000	6	0.500	**1.000**	0.750	0.750	0.800	**1.000**	0.286	**1.000**	0.571	**1.000**	0.800	**1.000**	**0.995**	**0.985**	0.541	0.991
2000	8	**1.000**	**1.000**	**1.000**	0.750	**1.000**	**1.000**	**1.000**	**1.000**	0.800	**1.000**	0.800	**1.000**	**1.000**	**0.980**	0.071	**1.000**
4000	2	0.021	0.375	0.000	0.000	0.000	0.000	0.020	**1.000**	0.018	0.625	0.000	0.000	**1.000**	**0.980**	0.992	0.980
4000	4	0.400	0.250	0.400	0.250	0.400	0.250	0.400	0.250	0.375	0.375	0.250	0.125	**1.000**	**0.980**	0.599	0.992
4000	6	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.200	0.375	0.000	0.000	**0.982**	**0.980**	0.750	0.977
4000	8	0.889	**1.000**	0.889	**1.000**	0.889	**1.000**	**1.000**	**1.000**	0.889	**1.000**	0.800	**1.000**	**1.000**	**0.980**	0.946	0.981
6000	2	0.027	0.846	0.000	0.000	0.000	0.000	0.014	0.154	0.032	0.231	0.053	0.077	**0.963**	**0.978**	**0.988**	**0.978**
6000	4	0.000	0.000	0.000	0.000	0.030	0.385	0.028	0.308	0.017	0.231	0.032	0.462	**0.785**	**0.979**	0.024	0.778
6000	6	0.393	0.917	0.636	0.583	0.571	0.667	0.417	0.833	0.588	0.833	0.529	0.750	0.998	**0.982**	**1.000**	**0.982**
6000	8	0.333	0.250	0.333	0.250	0.632	1.000	0.533	0.667	0.500	0.583	0.417	0.417	**1.000**	**0.980**	**0.966**	**0.984**
8000	2	0.018	0.235	0.000	0.000	0.167	0.059	0.026	0.353	0.018	0.471	0.000	0.000	**0.926**	**0.980**	**0.937**	**0.977**
8000	4	0.279	0.706	0.000	0.000	0.068	0.765	0.026	0.765	0.061	0.824	0.038	0.765	0.000	0.000	**0.920**	**0.996**
8000	6	0.308	0.706	0.583	0.412	0.318	0.412	0.205	0.529	0.185	0.706	0.263	0.294	**0.937**	**0.980**	0.024	**1.000**
8000	8	0.362	**1.000**	0.607	**1.000**	0.727	0.941	0.362	**1.000**	0.333	**1.000**	0.714	0.882	**0.999**	**0.979**	0.994	0.994
10000	2	0.027	0.524	0.000	0.000	0.008	0.048	0.022	1.000	0.017	0.714	0.000	0.000	0.596	0.977	**0.958**	**0.979**
10000	4	0.141	0.619	**1.000**	0.048	0.326	0.714	0.017	0.143	0.389	0.667	0.304	0.667	**0.991**	**0.987**	0.775	**0.995**
10000	6	0.316	0.571	0.333	0.381	0.356	0.762	0.297	0.524	0.277	0.619	0.297	0.524	**0.984**	**0.985**	0.756	**0.999**
10000	8	0.400	0.762	0.379	0.524	0.633	0.905	0.447	0.810	0.415	0.810	0.621	0.857	**1.000**	**0.979**	**0.994**	**0.985**
20000	2	0.027	0.595	0.000	0.000	0.026	0.238	0.021	0.976	0.021	**1.000**	0.011	0.071	**0.932**	**0.979**	0.654	**0.982**
20000	4	0.133	0.500	**1.000**	0.143	0.372	0.690	0.021	**1.000**	0.246	0.667	0.322	**0.929**	**0.971**	**0.987**	**0.925**	**0.995**
20000	6	0.287	0.595	0.405	0.405	0.300	0.429	0.245	0.595	0.291	0.595	0.302	0.381	**0.699**	**0.989**	0.322	**0.986**
20000	8	**1.000**	**0.952**	**1.000**	0.929	**1.000**	0.929	0.976	0.976	0.820	0.976	**1.000**	0.881	0.019	0.927	**0.962**	**0.995**

- for **10000 elements** and **four features** dataset is the **VQC model with basic entangling layer considering YXZ encoding and 8 layers without re-uploading** technique, providing a precision and a recall close to the best;
- for **10000 elements** and **six features** dataset is the **VQC model with basic entangling layer considering XYZ encoding and 2 layers with re-uploading** technique, providing the best precision and a recall close to the best;
- for **10000 elements** and **eight features** dataset is the **VQC model with basic entangling layer considering YX_H encoding and 4 layers with re-uploading** technique, providing the best precision and a recall close to the best;
- for **20000 elements** and **two features** dataset is the **VQC model with basic entangling layer considering YX_H encoding and 8 layers with re-uploading** technique, providing a precision and a recall close to the best;
- for **20000 elements** and **four features** dataset is the **VQC model with basic entangling layer considering ZYX_H encoding and 6 layers without re-uploading** technique, providing a precision and a recall close to the best;
- for **20000 elements** and **six features** dataset is the **VQC model with basic entangling layer considering Y_H encoding and 10 layers without re-uploading** technique, providing the best precision and recall;
- for **20000 elements** and **eight features** dataset is **logistic regression**, providing a precision and a recall close to the best.

F. IMPACT OF REAL-DEVICE NOISE

In order to estimate the effect of nowadays quantum computers' **noise** on developed QML models, we evaluate prediction quality of the **VQC model with Strongly Entangling Layer, considering XYZ encoding technique with re-uploading and 6 layers**, in **10000 transaction and six features validation dataset** of the on the **127-qubit** `ibm_brisbane` device submitting circuits via cloud with **IBM Quantum Runtime Services** (Pay-as-you-go plan of Intesa Sanpaolo). The obtained results are discussed in the following.

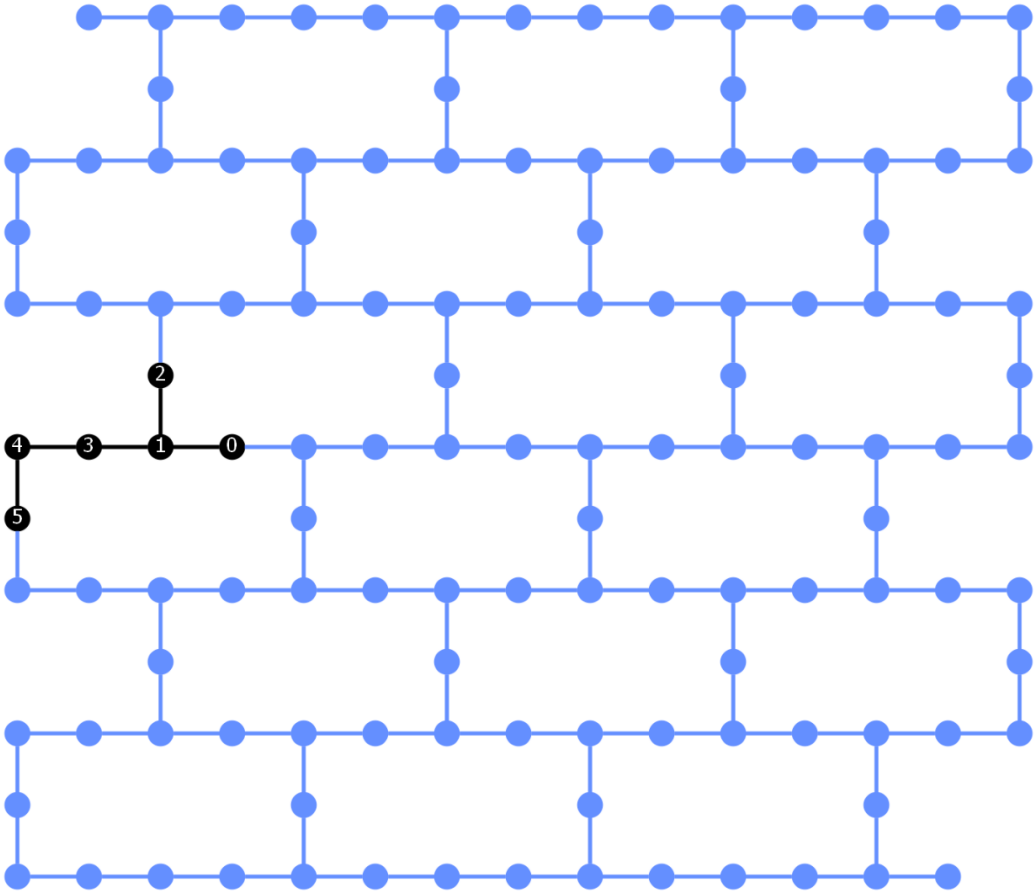
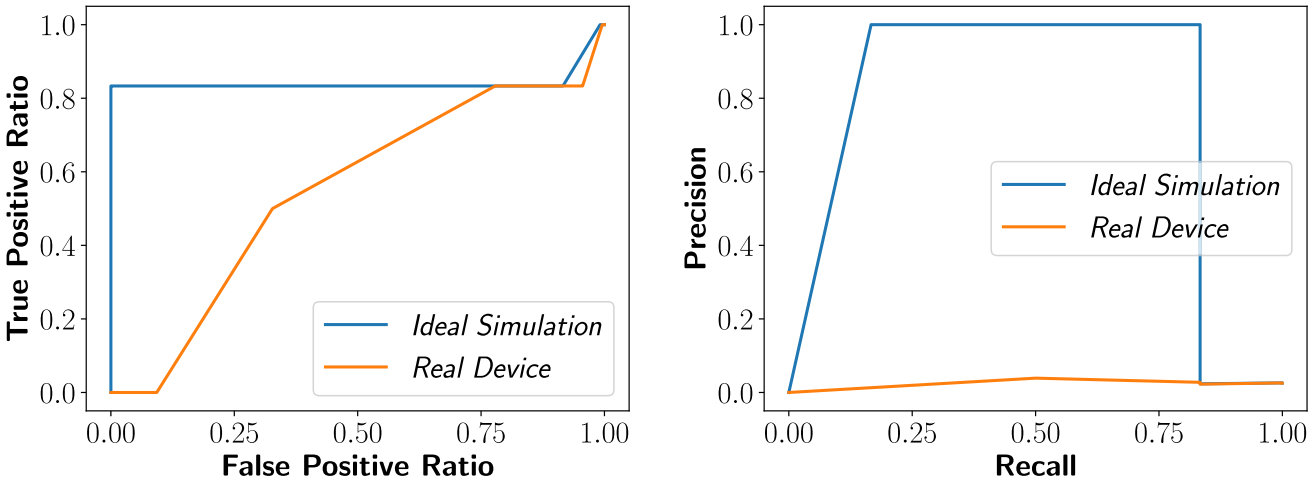


FIGURE 17. Quantum circuit mapped on the `ibm_brisbane` quantum computer.



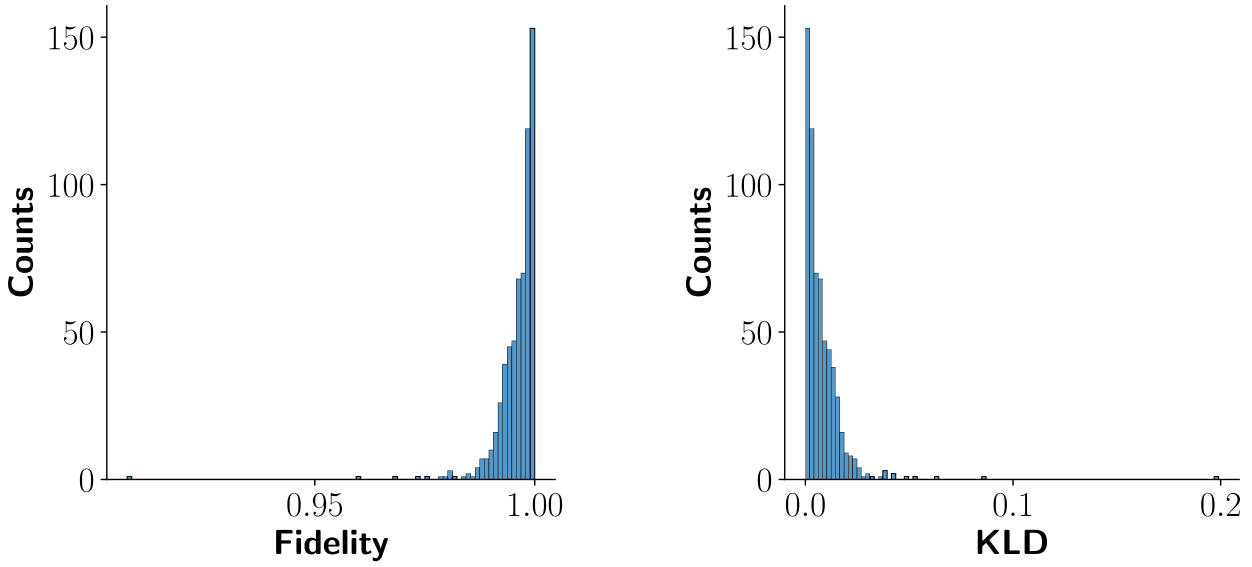
(a) ROC curve

(b) Precision-Recall curve

FIGURE 18. ROC and precision-recall curves obtained by predicting outcomes of 10000 transactions and 6 features validation dataset with the VQC model with strongly entangling layer, considering XYZ encoding technique with re-uploading and 6 layers on the 127-qubit `ibm_brisbane` device and ideal Pennylane simulator.

Figure 17 shows one of the submitted circuits mapped on the real device. It is possible to observe that the circuit is executed on close qubits to limit the impact of noise.

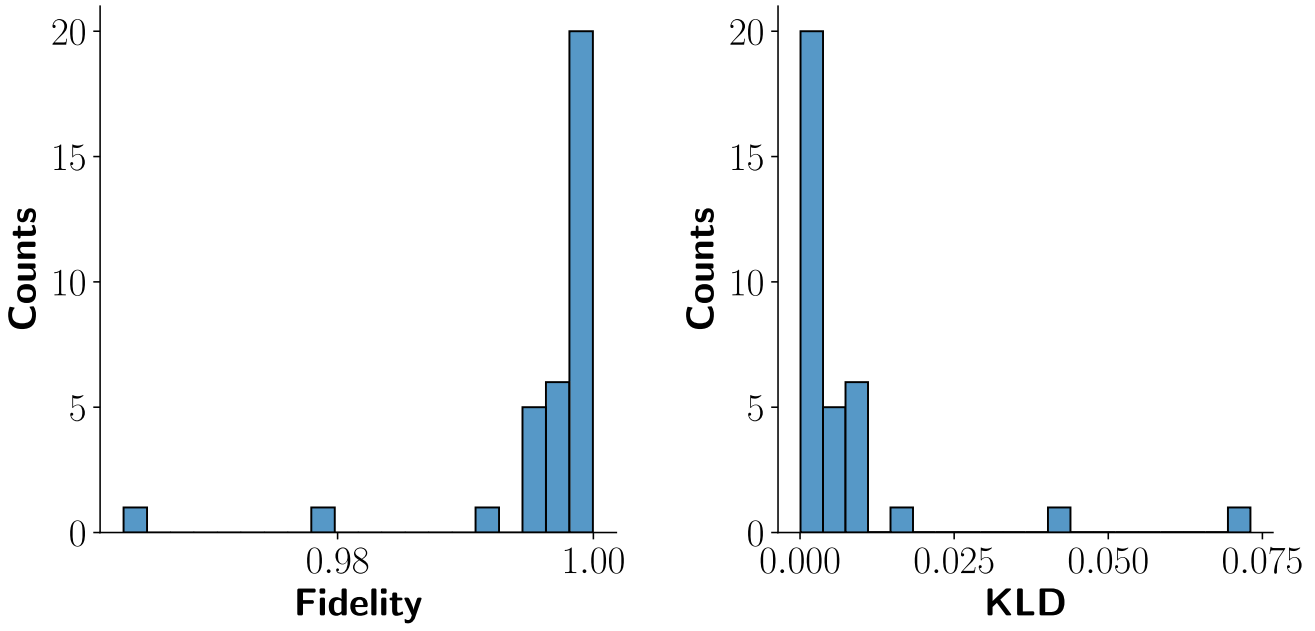
It is possible to observe from ROC and precision-recall curves reported in Figure 18, comparing ideal simulation and real device outcomes, that the **noise degrades completely the**



(a) Hellinger fidelity

(b) Kullback Leibler divergence (KLD)

FIGURE 19. Hellinger fidelity and KLD obtained comparing outcomes of `ibm_brisbane` device and ideal Pennylane simulator. The first shows the *closeness* among ideal and real results, while the second the *divergence*.



(a) Hellinger fidelity

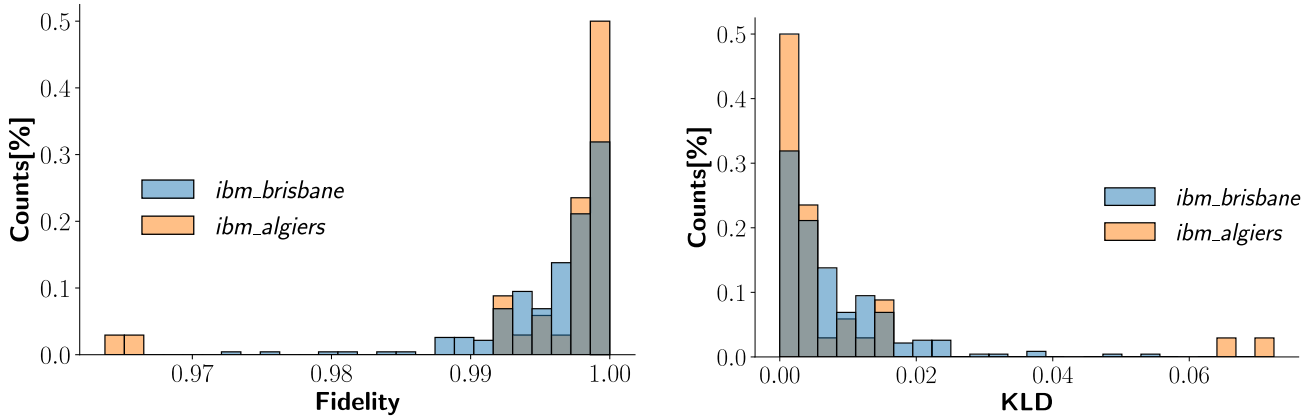
(b) Kullback Leibler divergence (KLD)

FIGURE 20. Hellinger fidelity and KLD obtained comparing outcomes of `ibm_algeris` device and ideal Pennylane simulator. The first shows the *closeness* among ideal and real results, while the second the *divergence*.

performance, especially in terms of precision-recall, proving that current device are not ready for employment in real application scenario.

To better estimate the impact of noise, **Hellinger fidelity** [31] ([0, 1]) and **Kullback Leibler divergence** (**KLD**) [32] divergence have been computed for all the performed test.

Fidelity is defined as a measure of the *closeness* of two quantum states, assuming value 1 if they are identical. In particular, it is described as $(1 - H^2)^2$, where H is the



(a) Fidelity

(b) Kullback Leibler divergence (KLD)

FIGURE 21. **Hellinger fidelity and KLD obtained comparing outcomes of `ibm_brisbane` device and `ibm_algeris` with ideal Pennylane simulator. The first shows the *closeness* among ideal and real results, while the second the *divergence*.**

Hellinger distance, computed as:

$$H(I,R) = \frac{1}{\sqrt{2}}\sqrt{\sum_{i=1}^{k}(\sqrt{I_i} - \sqrt{R_i})^2}\,, \tag{24}$$

where I and R are the probability distribution, i.e., the square module of the state vector, of the two quantum states, in this case, obtained from simulation and real device, respectively.

On the other hand, KLD estimates the *difference* between the two quantum states. Therefore, it is equal to 0 in the case of two identical states. The KLD is computed as:

$$D_{KL}(I\|R) = \sum_{x\in\chi} I(x)\log\left(\frac{I(x)}{R(x)}\right), \tag{25}$$

where I and R are the probability distribution, i.e. the square module of the state vector, of the two quantum states, in this case, obtained from simulation and real device, respectively.

The obtained fidelity and KLD distributions are shown in Figure 19. Even though the fidelity distribution is concentrated around the ideal value 1, the effect of the values spread is not negligible. Similarly, KLD presents a distribution close to 0, i.e., the ideal value, but its variance is relevant.

In order to evaluate the impact of the quantum device size on the obtained results, we evaluate the prediction outcomes on a subset of the validation dataset with the 27-qubit `ibm_algiers` computer. Fidelity and KLD divergence obtained are shown in Figure 20. Even if also in this case the device behaviour is far from the ideal one, the outcomes are more accurate than ones obtained with `ibm_brisbane`, as shown in the comparative Figure 21.

G. DISCUSSION

The exploration presented in this article proves that **quantum models are a promising alternative to classical ones** in some critical applications like fraud detections. In particular, they allow **the achievement of better results in terms of precision and recall**, which are the most relevant figures of merit in this kind of application. However, **quantum devices still need to be much improved** before being competitive and comparable with classic solutions. Indeed, the encouraging results obtained with the ideal simulator have not been found by performing tests on real devices.

Moreover, it is important to notice that, on average, among the considered quantum models, those with the **most complex encoding mechanism perform better** than the others.

The operative methodology presented in this article is generic and can be applied for any other applications involving an unbalanced dataset with minimal adjustment. However, one strength point of this work is that the employed **dataset consists of real bank transaction**, showing the ideal performance of quantum computers in a real-world scenario.

VI. CONCLUSION

This work broadly investigates the potential of the VQC-based quantum classification for fraud detection tasks. The analysis is conducted on **a dataset based on real Intesa Sanpaolo Bank transactions** properly reduced through **PCA** and sampling techniques to limit the complexity of the required quantum simulations. Results obtained from quantum models are **benchmarked against well-established classical methods**, including Logistic Regression, XGBoost, Support Vector Machine, Random Forest, and Neural Networks. This work proves that **quantum models provide advantages in terms of precision and recall with respect to classical models**, especially in datasets with a reduced number of features and elements. This can be seen as a trace of their potential in separating elements of a highly unbalanced dataset.

It has been proved that, on average, the **most complex encoding mechanism for VQC provides better results**.

The impact of noise has also been evaluated by performing predictions on `ibm_brisbane` quantum device, proving

that **current computers not yet ready to deal with real-world problems**. Indeed, non-idealities significantly compromise the results.

The trained quantum models are available on the Github repository. This way, transfer learning techniques like that treated in [33] can be exploited to adapt them to other datasets with similar characteristics.

Even though the current analysis mainly focuses on the VQC model considering the angle encoding mechanism, this can be considered a milestone for proving the exploitability of quantum models for real-world machine-learning tasks. First of all, alternative methods to PCA can be investigated for features reduction. For example, [34] proposes to perform this with tensor-train network (TTN) technique. Moreover, the analysis can be further expanded by considering **other encoding mechanisms**, such as amplitude one, **other ansatz circuits** [35], and other **quantum circuit model solutions**, like Quantum Support Vector machines (QSVM) [36] and of hybrid quantum-classical models as those proposed in [37]. Finally, machine learning procedures exploiting quantum computation in the parameter optimization part through proper **QUBO models** [38] can be explored for evaluating the potential benefits.

We hope that this analysis can encourage and support the exploration of quantum solutions for further classification tasks, positively impacting various aspects of human life.

ACKNOWLEDGMENT

The authors would like to thank Intesa Sanpaolo Quantum Competence Center for granting access to IBM Quantum Runtime Services (Pay-as-you-go plan) and HPC@POLITO—a project of Academic Computing within the Department of Control and Computer Engineering, Politecnico di Torino (http://hpc.polito.it)—for providing computational resources.

Moreover, the authors would like to thank Intesa Sanpaolo team of cybersecurity, anti-fraud and data analytic solutions for providing the real dataset employed in the analysis through an agreement between the Politecnico di Torino and Intesa Sanpaolo. The datasets analyzed during the current study are available from Intesa Sanpaolo on reasonable request and after finding a proper agreement.

REFERENCES

[1] Y. Kou, C.-T. Lu, S. Sirwongwattana, and Y.-P. Huang, "Survey of fraud detection techniques," in *Proc. IEEE Int. Conf. Netw., Sens. Control*, Mar. 2004, pp. 749–754, doi: 10.1109/ICNSC.2004.1297040.

[2] P. Tiwari, S. Mehta, N. Sakhuja, J. Kumar, and A. K. Singh, "Credit card fraud detection using machine learning: A study," 2021, *arXiv:2108.10005*.

[3] N. Boutaher, A. Elomri, N. Abghour, K. Moussaid, and M. Rida, "A review of credit card fraud detection using machine learning techniques," in *Proc. 5th Int. Conf. Cloud Comput. Artif. Intell., Technol. Appl. (CloudTech)*, Nov. 2020, pp. 1–5, doi: 10.1109/CloudTech49835.2020.9365916.

[4] A. Thennakoon, C. Bhagyani, S. Premadasa, S. Mihiranga, and N. Kuruwitaarachchi, "Real-time credit card fraud detection using machine learning," in *Proc. 9th Int. Conf. Cloud Comput., Data Sci. Eng. (Confluence)*, Jan. 2019, pp. 488–493, doi: 10.1109/CONFLUENCE.2019.8776942.

[5] J. O. Awoyemi, A. O. Adetunmbi, and S. A. Oluwadare, "Credit card fraud detection using machine learning techniques: A comparative analysis," in *Proc. Int. Conf. Comput. Netw. Informat. (ICCNI)*, Oct. 2017, pp. 1–9, doi: 10.1109/ICCNI.2017.8123782.

[6] E. Zahedinejad and A. Zaribafiyan, "Combinatorial optimization on gate model quantum computers: A survey," 2017, *arXiv:1708.05294*.

[7] D. Herman, C. Googin, X. Liu, A. Galda, I. Safro, Y. Sun, M. Pistoia, and Y. Alexeev, "A survey of quantum computing for finance," 2022, *arXiv:2201.02773*.

[8] H. Wang, W. Wang, Y. Liu, and B. Alidaee, "Integrating machine learning algorithms with quantum annealing solvers for online fraud detection," *IEEE Access*, vol. 10, pp. 75908–75917, 2022, doi: 10.1109/ACCESS.2022.3190897.

[9] J. Mancilla and C. Pere, "A preprocessing perspective for quantum machine learning classification advantage in finance using NISQ algorithms," *Entropy*, vol. 24, no. 11, p. 1656, Nov. 2022, doi: 10.3390/e24111656.

[10] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information*, 10th ed., Cambridge, U.K.: Cambridge Univ. Press, 2010.

[11] S. Raschka and V. Mirjalili, *Python Machine Learning: Machine Learning and Deep Learning With Python, Scikit-Learn, and TensorFlow 2*. Packt, 2019.

[12] E. Combarro, S. Gonzalez-Castillo, and A. Di Meglio, *A Practical Guide to Quantum Machine Learning and Quantum Optimization: Hands-On Approach to Modern Quantum Algorithms*. Packt, 2023.

[13] M. Schuld and F. Petruccione, *Machine Learning With Quantum Computers*. Springer, 2021.

[14] T. Kurita, "Principal component analysis (PCA)," in *Computer Vision: A Reference Guide*. Springer, 2020, pp. 1–4, doi: 10.1007/978-3-030-03243-2_649-1.

[15] R. Xu and D. WunschII, "Survey of clustering algorithms," *IEEE Trans. Neural Netw.*, vol. 16, no. 3, pp. 645–678, May 2005, doi: 10.1109/TNN.2005.845141.

[16] D. G. Kleinbaum, K. Dietz, M. Gail, M. Klein, and M. Klein, *Logistic Regression*. Springer, 2002.

[17] A. Parmar, R. Katariya, and V. Patel, "A review on random forest: An ensemble classifier," in *Proc. Int. Conf. Intell. Data Commun. Technol. Internet Things*. Cham, Switzerland: Springer, 2019, pp. 758–763, doi: 10.1007/978-3-030-03146-6_8.

[18] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2016, pp. 785–794, doi: 10.1145/2939672.2939785.

[19] S. Dreiseitl and L. Ohno-Machado, "Logistic regression and artificial neural network classification models: A methodology review," *J. Biomed. Informat.*, vol. 35, nos. 5–6, pp. 352–359, Oct. 2002, doi: 10.1016/s1532-0464(03)00034-0.

[20] J. Cervantes, F. Garcia-Lamont, L. Rodríguez-Mazahua, and A. Lopez, "A comprehensive survey on support vector machine classification: Applications, challenges and trends," *Neurocomputing*, vol. 408, pp. 189–215, Sep. 2020, doi: 10.1016/j.neucom.2019.10.118.

[21] N. Innan, A. Sawaika, A. Dhor, S. Dutta, S. Thota, H. Gokal, N. Patel, M. A.-Z. Khan, I. Theodonis, and M. Bennai, "Financial fraud detection using quantum graph neural networks," *Quantum Mach. Intell.*, vol. 6, no. 1, pp. 1–18, Jun. 2024, doi: 10.1007/s42484-024-00143-6.

[22] T. Priyaradhikadevi, S. Vanakovarayan, E. Praveena, V. Mathavan, S. Prasanna, and K. Madhan, "Credit card fraud detection using machine learning based on support vector machine," in *Proc. 8th Int. Conf. Sci. Technol. Eng. Math. (ICONSTEM)*, Apr. 2023, pp. 1–6, doi: 10.1109/ICONSTEM56934.2023.10142247.

[23] W. Li, Z.-D. Lu, and D.-L. Deng, "Quantum neural network classifiers: A tutorial," *SciPost Phys. Lect. Notes*, p. 061, Aug. 2022, doi: 10.21468/scipostphyslectnotes.61.

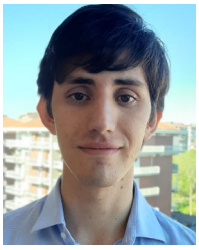
[24] A. Tudisco, "Encoding techniques for quantum machine learning," M.S. thesis, Politecnico di Torino, 2022. [Online]. Available: https://webthesis.biblio.polito.it/25437/

[25] V. Bergholm et al., "PennyLane: Automatic differentiation of hybrid quantum-classical computations," 2018, *arXiv:1811.04968*.

[26] M. Kölle, A. Giovagnoli, J. Stein, M. B. Mansky, J. Hager, T. Rohe, R. Müller, and C. Linnhoff-Popien, "Weight re-mapping for variational quantum algorithms," 2023, *arXiv:2306.05776*.

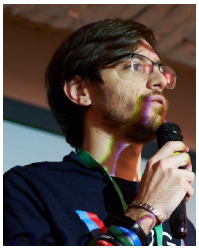
[27] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016. [Online]. Available: http://www.deeplearningbook.org

[28] M. Baak, R. Koopman, H. Snoek, and S. Klous, "A new correlation coefficient between categorical, ordinal and interval variables with Pearson characteristics," *Comput. Statist. Data Anal.*, vol. 152, Dec. 2020, Art. no. 107043, doi: 10.1016/j.csda.2020.107043.
[29] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*.
[30] *Intel Xeon Gold 6134 Processor—Product Specification*. Accessed: Oct. 25, 2021. [Online]. Available: https://ark.intel.com/content/www/us/en/ark/products/120493/intel-xeon-gold-6134-processor-24-75m-cache-3-20-ghz.html
[31] *Qiskit 0.36.2 Documentation*. Accessed: Apr. 4, 2024. [Online]. Available: https://tinyurl.com/mr27xdmc
[32] S. Kullback and R. A. Leibler, "On information and sufficiency," *Ann. Math. Statist.*, vol. 22, no. 1, pp. 79–86, Mar. 1951, doi: 10.1214/aoms/1177729694.
[33] J. Qi and J. Tejedor, "Classical-to-quantum transfer learning for spoken command recognition based on quantum neural networks," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2022, pp. 8627–8631.
[34] J. Qi, C.-H.-H. Yang, P.-Y. Chen, and M.-H. Hsieh, "Theoretical error performance analysis for variational quantum circuit based functional regression," *npj Quantum Inf.*, vol. 9, no. 1, p. 4, Jan. 2023.
[35] Z. Holmes, K. Sharma, M. Cerezo, and P. J. Coles, "Connecting ansatz expressibility to gradient magnitudes and barren plateaus," *PRX Quantum*, vol. 3, no. 1, Jan. 2022, Art. no. 010313, doi: 10.1103/prxquantum.3.010313.
[36] S. Kavitha and N. Kaulgud, "Quantum machine learning for support vector machine classification," *Evolutionary Intell.*, vol. 17, pp. 819–828, Jul. 2022, doi: 10.1007/s12065-022-00756-5.
[37] C. H. Yang, J. Qi, S. Y. Chen, Y. Tsao, and P.-Y. Chen, "When BERT meets quantum temporal convolution learning for text classification in heterogeneous computing," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2022, pp. 8602–8606.
[38] P. Date, D. Arthur, and L. Pusey-Nazzaro, "QUBO formulations for training machine learning models," *Sci. Rep.*, vol. 11, no. 1, p. 10029, May 2021, doi: 10.1038/s41598-021-89461-4.



ANTONIO TUDISCO (Graduate Student Member, IEEE) received the B.Sc. and M.Sc. degrees in electronic engineering from the Politecnico di Torino, Italy, in 2020 and 2022, respectively, where he is currently pursuing the Ph.D. degree. His research interests include quantum computing, with a particular focus on the development of quantum machine learning models.

DEBORAH VOLPE (Graduate Student Member, IEEE) received the B.Sc. and M.Sc. degrees in electronic engineering from the Politecnico di Torino, in 2019 and 2021, respectively, where she is currently pursuing the Ph.D. degree in electrical, electronics, and communications engineering. Her research interests include the emulation of quantum computers on classical hardware (FPGA, CPU, and GPU), quantum-compliant approaches for solving QUBO problems, and quantum applications, in particular QUBO formulations.



GIACOMO RANIERI received the bachelor's degree in electronics, computer science and telecommunications engineering in Bologna, in 2015, and the master's degree in computer science engineering, in 2018. In 2022, he joined the Quantum Computing Competence Center, Intesa Sanpaolo, as a Quantum Software Engineer. This role allows him to pursue his passion for cutting-edge technologies.

GIANBIAGIO CURATO received the B.Sc. and M.Sc. degrees in physics from the University of Florence, in 2007 and 2010, respectively, and the Ph.D. degree in financial mathematics from the Scuola Normale Superiore of Pisa, in 2015. He is currently with Intesa Sanpaolo, Cybersecurity Antifraud Solution Team. His research interests include data analytics and machine learning projects applied to financial fraud detection.

DAVIDE RICOSSA received the master's degree in mathematics from the University of Turin, with a thesis focused on stochastic signal processing. In 2019, he commenced his career as a Quantitative Developer with Intesa Sanpaolo, where he later began researching the financial applications of quantum computing. His research interests include quantitative finance, quantum algorithm design, stochastic differential equations, and optimal stopping problems.

MARIAGRAZIA GRAZIANO received the M.Sc. and Ph.D. degrees in electronics engineering from the Politecnico di Torino, Turin, Italy, in 1997 and 2001, respectively. Since 2008, she has been an Adjunct Faculty Member with the University of Illinois at Chicago (UFL), Chicago, IL, USA. From 2014 to 2017, she was a Marie-SklodowskaCurie IntraEuropean Fellow with London Centre for Nanotechnology, University College London, U.K. Since 2015, she has been a Lecturer with the École Polytechnique Fédérale de Lausanne, Switzerland. She is currently an Associate Professor with the Department of Applied Science and Technology, Politecnico di Torino. Her research interests include nanotechnology, emerging technology devices, the simulation and design of nanocomputing devices, architectures, and circuits with physical-aware computer-aided design (CAD) tools, field coupling nanocomputing, and quantum computing.

DAVIDE CORBELLETTO was born in Italy. He is currently a Mathematician with almost 20 years of professional experience in computer science. Before joining Intesa Sanpaolo, in 2016, he previously worked for almost 12 years as a Technical Advisor for several international firms, banks, and insurance companies. He has developed strong capabilities in managing digital transformation projects mainly focused on designing reliable data architecture, promoting cloud computing adoption, and conceiving scalable machine learning solutions. Since 2020, he has been dealing with quantum computing within the brand-new dedicated Competence Center, Group Technology Area.

Open Access funding provided by 'Politecnico di Torino' within the CRUI CARE Agreement