Eur. Phys. J. B (2024)97:178
https://doi.org/10.1140/epjb/s10051-024-00786-1

THE EUROPEAN PHYSICAL JOURNAL B

Regular Article - Computational Methods

# Synthetic data generation with hybrid quantum-classical models for the financial sector

Otto M. Pires[1,a], Mauro Q. Nooblath[1,b], Yan Alef C. Silva[1,c], Maria Heloísa F. da Silva[1,2,d], Lucas Q. Galvão[1,e], and Anton S. Albino[1,f]

[1] Latin America Quantum Computing Center, SENAI-CIMATEC, Av. Orlando Gomes, 1845, Salvador, Bahia 41650-010, Brazil
[2] Universidade Federal do Oeste da Bahia - Campus Reitor Edgard Santos, UFOB, Rua Bertioga, 892, Morada Nobre I, Barreiras, Bahia 47810-059, Brazil

Received 26 February 2024 / Accepted 10 September 2024


**Abstract.** Data integrity and privacy are critical concerns in the financial sector. Traditional methods of data collection face challenges due to privacy regulations and time-consuming anonymization processes. In collaboration with Banco BV, we trained a hybrid quantum-classical generative adversarial network (HQGAN), where a quantum circuit serves as the generator and a classical neural network acts as the discriminator, to generate synthetic financial data efficiently and securely. We compared our proposed HQGAN model with a fully classical GAN by evaluating loss convergence and the MSE distance between the synthetic and real data. Although initially promising, our evaluation revealed that HQGAN failed to achieve the necessary accuracy to understand the intricate patterns in financial data. This outcome underscores the current limitations of quantum-inspired methods in handling the complexities of financial datasets.

## 1 Introduction

Data is essential for artificial intelligence development, increasing the demand for high-quality data [1–3]. However, stricter data privacy regulations have made the collection and labeling of real data more difficult [4–6]. In the financial sector, sensitive customer data is used for software testing and fraud detection, necessitating strict data management protocols. This can delay access to anonymized data, introducing errors and compromising quality. Synthetic data offers a solution by protecting training data, as it only reveals synthetic variants in case of security breaches [7,8]. It is also cheaper to produce, automatically labeled, and avoids many privacy issues [9]. The challenge is to generate synthetic financial datasets that maintain the statistical characteristics of the original data without traceability to individuals.

Generative Adversarial Networks (GANs) [10] stand out as a powerful tool for data generation. They have gained significant traction in the field of generative learning and find application in an extensive variety of domains [11–13]. In the finance domain, applications of GANs include financial data generation [12,14,15], stock market prediction [16,17], credit scoring [18] and fraud detection [19,20]. In the image processing domain, GANs are used notably for image super-resolution (ISR) [21,22] that can also improve early medical diagnosis in clinical pathology [23] to name a few. Furthermore, GANs have been applied for tasks such as protein engineering sequential data based applications [24], molecule development [25], drug discovery [26], music genre fusion and music generation [27], video generation and prediction [28], autonomous driving [29], weather forecasting [30], astronomy imaging [31], among others. The goal of GANs is to simultaneously train two neural networks: a generator G, and a discriminator D, through an adversarial learning strategy [32].

At present, quantum computers hold the promise of tackling problems that are deemed unsolvable by classical computers. Moreover, there is ongoing progress in extending these concepts and algorithms into the domain of quantum machine learning [33]. Recent theoretical studies indicate that quantum generative models might possess an exponential advantage over their classical counterparts [23,34,35]. However, a pivotal question that requires attention in the field of quantum GAN is whether existing quantum devices possess the capability for real-world generative learning. This

[a] e-mail: otto.pires@fieb.org.br (corresponding author)
[b] e-mail: mauro.neto@fieb.org.br
[c] e-mail: yan.chagas@fieb.org.br
[d] e-mail: maria.fraga@fbter.org.br
[e] e-mail: lucas.queiroz@fbter.org.br
[f] e-mail: anton.albino@fieb.org.br

Published online: 18 November 2024

capacity is directly linked to the practical application of quantum GANs on near-term quantum devices [36].

We propose a hybrid model that uses a quantum circuit as generator and a classical neural network as discriminator following the model proposed in [36]. Our main objective is the generation of synthetic data related to the approval of money withdraw from Banco BV, a brazilian bank. For precise evaluation of our generative models, we employ the Mean Squared Error (MSE) as a quantitative distance metrics to assess the disparity between real and generated distributions.

Therefore, the primary novel aspects and contributions of this research to the contemporary state of the art in this field are outlined as follows:

1. The proposal, development and investigation of a hybrid quantum-classical generative adversarial network model for synthetic financial data generation;
2. A comprehensive examination and comparison of the effectiveness of the proposed model in contrast to a conventional classical GAN, evaluating both using the loss function convergence and the MSE distance between the synthetic data generated by both models and the original dataset;
3. The application of the proposed quantum model and the baseline model in a case study scenario utilizing real financial data offered by Banco BV.

This work is organized as follows. First, in Sect. 2, we present the necessary information to understand GANs, complemented in Sect. 2.1 which describes the quantum version of the model. We describe our proposed solution in Sect. 3. In Sect. 4, we describe our experiments and analyse the results. Finally, we close with a conclusion and future works in Sect. 5.

## 2 Generative adversarial networks

A Generative Adversarial Network is a deep learning technique introduced by Goodfellow et al. [10], which aims to generate synthetic data from a given dataset using a pair of neural networks, namely the discriminator and the generator. The discriminator network learns to distinguish between real and synthetic data, while the generator network learns to produce synthetic data that closely resembles the real data. Through an adversarial training process, the two networks engage in a competitive interplay two-player minmax game, continually improving their performance iteratively. This process aims to reach a Nash equilibrium, where the generator creates high-quality synthetic data that is indistinguishable from real data [37]. However, this point is rarely achieved in practice apart from the most simple tasks [38].

The adversarial learning process starts by defining a random distribution represented as a latent vector $z$ sampled from a probability distribution $p(z)$, such as a uniform or Gaussian distribution. As the term implies, the latent vector is defined in the latent space: a space proficient in compactly representing data [39]. Thus, instead of dealing with a more complex space, such as the data space, the latent space is a manageable alternative, once the data representation is packed.

The generator then maps this latent vector to the data space, generating synthetic samples denoted as $G(z)$, where $G$ represents the mapping from the latent space to the data space. On the other hand, the discriminator network takes both real samples from the dataset and synthetic samples generated by the generator as inputs. Its objective is to distinguish between real and synthetic samples and assign a probability value indicating the likelihood of each sample as being real. If the probability value is 1, then the data is taken as real; if it is 0 the data is taken as fake. The expected probability value is 0.5, which indicates that the discriminator is unable to differentiate fake and real samples. In mathematical notation, we can write the discriminator function $D(x)$ as the probability of the discriminator to classify the real data as real and $D(G(z))$ as the probability of the discriminator to classify the synthetic data as real.

An efficacious approach to evaluate these probability values is using functions able to measure the discrepancy between the real and synthetic data, also called objective functions [37]. Some authors have proposed different models for these functions such as *f-Divergence* to quantify the differences between two distributions with a particular convex function, *Integral Probability Metric (IPM)* to get the maximal measure between two arbitrary distributions, and *Auxiliary Object Functions* for both reconstruction object function and classification object functions [40]. In the original GANs paper, Goodfellow et al. [10] formulated the following objective function:

$$\min_G \max_D V(D,G) = E_{x\sim p_{data}(x)}[log(D(x))] + E_{z\sim p_g(z)}[log(1-D(G(z)))] \quad (1)$$

where $G$ must be trained simultaneously to minimize $log(1-D(G(z)))$.

Currently, there are a notable number of GAN variants, which require particular objective functions. Among these variations, some have gained prominence, such as *Deep Convolutional GAN* (DCGAN), in which introduces convolutional layers for learning hierarchical features [41]; *Conditional GAN* (CGAN), extended to a conditional model in the event that the generator and discriminator networks are dependent on additional information [42]; *Bidirectional GAN* (BiGAN), in which learns the inverse mapping and semantics of data distribution [43]; and others. A comparative summary of GAN variants can be found in reference [44].

### 2.1 Quantum GAN

The initial idea of a quantum GAN arose from Goodfellow's proposal, which was based on classical GANs [11]. A quantum GAN uses the same training principle as a

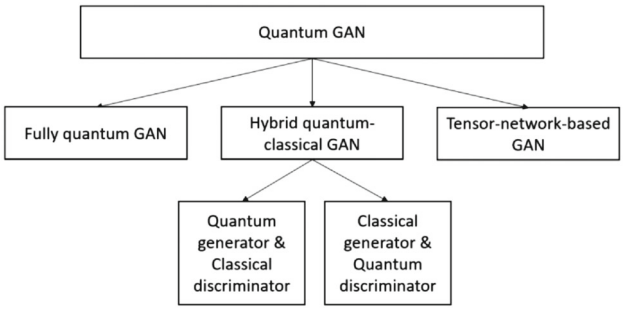


**Fig. 1** Quantum GAN structures

GAN, where it has a generator G and a discriminator D which are adversaries and can be fully quantum, hybrid or based on tensor networks [11] the scheme (Fig. 1) shows how they can be characterized.

In the case of a fully quantum system, both the generator and the discriminator are quantum. A minimum margin of error will always be associated with them, due to operators having a norm greater than or equal to 1.

The GAN models known as Tensor Network are based on a tensor that describes the quantum wave function in a multi-partite system, reducing it to tensors of lower rank [11]. This model boasts a distinctive advantage: it establishes a unified framework enabling both quantum and classical computing to leverage identical development techniques. Tensor Network models are trained in classical environments, but it is possible to transfer them to a quantum environment without the need for modifications. Their use is feasible in environments that handle both classical and quantum data, making them ideal for assisted machine learning applications [45].

And the third type are hybrid GANs, where we can have two configurations, the classical dataset generator and the quantum discriminator, a detail that we must pay attention to in this case is that the classical generator can never generate the statistics of a set of data similar to those of a source of quantum data [46]. Therefore, the discriminator will act in this situation, similar to the fully quantum case, and will always find a measurement smaller than $\frac{1}{2}$, and will never reach the Nash equilibrium [34].

The second type of hybrid GANs are quantum dataset generators and classical discriminators. This model is the most used in hybrid GANs, as in the opposite situation it is impossible for the generator to be able to estimate and learn a distribution that can defeat the discriminator. Therefore, the hybrid GAN model adopted in this work provides better performance than compared to traditional GANs. Another advantage is that, as the discriminator is classic, data coding is not necessary. In the Fig. 2 we can see a schematic of how the hybrid GAN works.

## 3 Method

In this research, we focus on the application of a Hybrid Quantum-Classical Generative Adversarial Network (HQGAN) for the generation of financial data pertinent to the approval of money withdraw by a Brazilian bank.

### 3.1 Case study

The dataset used in our study is provided by Banco BV and encompasses the profiles of bank clients who request to withdraw money from the Fundo de Garantia do Tempo de Serviço (FGTS) (English: Length-of-Service Guarantee Fund) [47]. The FGTS is a fund established to protect workers in Brazil who are dismissed without just cause. The data spans from April 2022 to July 2022 and includes over 13 million samples. Among these samples, only 54,490 (approximately 0.4%) represent clients classified as eligible to withdraw money from the FGTS.

Given the significant class imbalance in the dataset, conventional machine learning models face challenges in accurately learning the patterns needed to predict eligibility. The issue of imbalanced datasets can be addressed by Downsampling and Upweighting techniques [48], but even so they have limited capabilities in extreme imbalanced data, where the minority class is $< 1\%$ of the data set. To address this issue, we use synthetic data generation to increase the number of samples from the minority class (approved clients) to create a more balanced dataset for training future classification models.

The dataset contains 16 features, two of which are initially discarded. The discarded features are the target value, indicating whether the client is approved (1) or not (0), and the time feature, indicating the period of the request. We exclude the target feature because all synthetic data generated would have the same value (1) for this feature. We also consider the time period of the request irrelevant to the classification problem.

Preprocessing is essential for our quantum model. Only one feature in the dataset is categorical, so we use an ordinal encoder to convert its values into integers. Subsequently, we normalize all features by scaling them within a range of 0 to 1 using the min-max scaler procedure. In min-max normalization, each data point is adjusted to lie between 0 and 1:

$$x' = \frac{x - x_{\min}}{x_{max} - x_{\min}} \tag{2}$$

where $x'$ is the normalized data set and $x_{\min}$ and $x_{max}$ are the minimum and maximum values of the original data $x$, respectively.

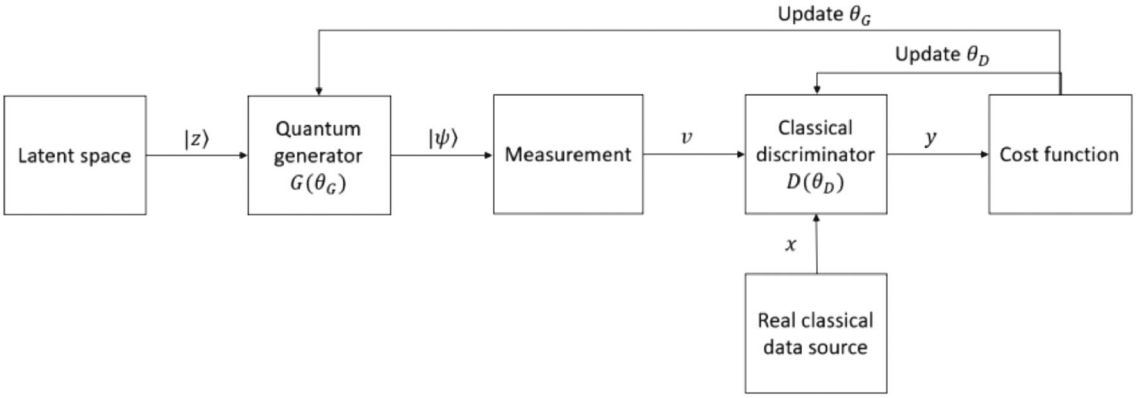


**Fig. 2** Hybrid quantum-classical GAN

### 3.2 Proposed method

Our proposed HQGAN architecture consists of a quantum generator paired with a classical discriminator. Our quantum generator is inspired by the resource-efficient implementation proposed by Huang et al. [36], which is detailed at [38]. This implementation, known as the Quantum Patch GAN, employs multiple quantum generators to construct small patches of the final data, which are subsequently concatenated to form the complete dataset. However, given the relatively small number of features in our dataset, the patch strategy was deemed unnecessary, resulting in a final generator with a single sub-generator. The overall quantum generator consists of $N$ qubits, which are decomposed into two parts, where the first $N_G$ data qubits are used to generate feature vectors and the remaining $N_A$ ancilla qubits are used to introduce nonlinearity [36].

One constraint of our proposed model is that the quantum generator's capacity to generate data points is contingent upon the number of data qubits $N_G$, specifically constrained to $2^{N_G}$. Consequently, to accommodate this constraint, we padded our dataset with two additional features initialized to 0, resulting in a total of 16 features for $N_G = 4$ and $N_A = 1$.

The quantum generator in our HQGAN is based on a Quantum Neural Network (QNN) model, consisting of a feature map and parameterized quantum circuits, also known as an ansatz, as depicted in Fig. 3. Our feature map includes an angle embedding circuit with $R_y$ rotations, specifically designed to encode classical data into a quantum state by representing each feature with a corresponding rotation angle applied to a qubit.

The ansatz in our model utilizes the same circuit structure as the angle embedding feature map across its layers, enhanced by a sequence of controlled-Z operations to induce entanglement. We chose a linear entanglement scheme, employing controlled-Z gates between neighboring qubits arranged in a linear configuration to generate entanglement [49]. The structure of the ansatz can be repeated to create deeper circuits and to improve the learning capabilities of the model.

By incorporating these components, our quantum generator efficiently encodes and processes classical data, leveraging the advantages of quantum computing to enhance the data generation capabilities of our HQGAN architecture.

A recognized limitation in employing a quantum model for synthesizing data is rooted in the quantum circuit model, where quantum gates operate as unitary transformations, inherently linear in nature. To enable the quantum generator to perform more complex generative tasks, non-linear transformations are essential. This can be achieved through the use of partial measurements and ancilla qubits [36]. For a given generator, the pre-measurement quantum state is represented as

$$|\psi(z)\rangle = U_G(\theta)|z\rangle,$$

where $U_G$ denotes the overall unitary of the generator and $|z\rangle$ is a quantum state encoding the latent vector. Upon a partial measurement $\Pi$ and tracing out the ancillary subsystem, $\mathcal{A}$,

$$\begin{aligned}\rho(z) &= \frac{\mathrm{Tr}_{\mathcal{A}}(\Pi \otimes \mathbb{I}|\psi(z)\rangle\langle\psi(z)|)}{\mathrm{Tr}(\Pi \otimes \mathbb{I}|\psi(z)\rangle\langle\psi(z)|} \\ &= \frac{\mathrm{Tr}_{\mathcal{A}}(\Pi \otimes \mathbb{I}|\psi(z)\rangle\langle\psi(z)|)}{\langle\psi(z)|\Pi \otimes \mathbb{I}|\psi(z)\rangle}\end{aligned}$$

the resultant post-measurement state, $\rho(z)$, exhibits non-linear transformation characteristics dependent on the input latent vector $z$.

The discriminator employed in our HQGAN is a neural network with a single hidden layer. The detailed architecture of this network is shown in Table 1. Our discriminator utilizes a straightforward linear architecture, featuring ReLU activation functions for the input and hidden layers, and a sigmoid activation function for the output layer. This configuration is standard for binary classification tasks, where the sigmoid function at the output layer provides a probability score, indicating the likelihood that the input belongs to a particular class.

The linear layers in the network apply a linear transformation to the input data. As described in the

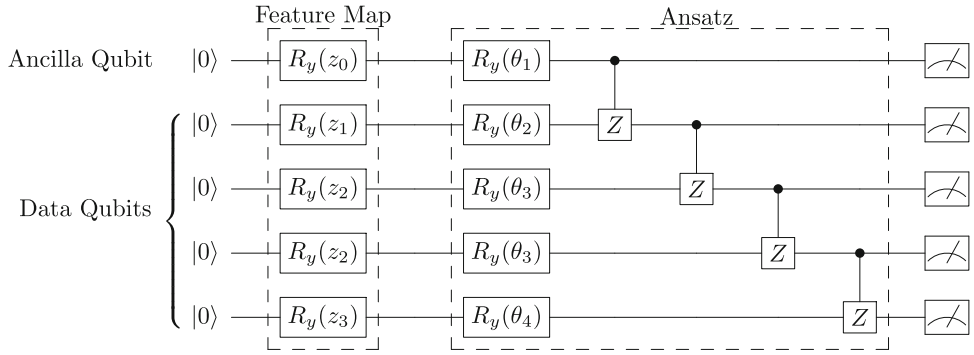


**Fig. 3** Quantum circuit design for a quantum generator used in the proposed HQGAN architecture. The circuit is divided into two main sections: the Feature Map and the Ansatz. The feature map is implemented using an angle embedding circuit with $R_y$ rotations, which are applied to both the data qubits and an ancilla qubit. The ansatz is a parameterized quantum circuit defined by independent parameter vector $\boldsymbol{\theta}$, that utilizes the same circuit structure as the angle embedding feature map across its layers, enhanced by a sequence of controlled-Z operations to induce entanglement. The figure depicts a circuit with depth $k = 1$, deeper circuits would repeat the structure of the ansatz, adding new trainable parameters to the circuit

PyTorch documentation [50], this transformation follows the rule:

$$y = xA^T + b$$

where $y$ is the output, $x$ is the input, $A$ is the weight matrix, and $b$ is the bias term. This transformation is essential for mapping the input features to the next layer in the network, enabling the model to learn and make accurate predictions.

### 3.3 Experimental design

The evaluation of our proposed model's performance involves comparing it with a fully classical GAN. Our implementation is based on the Deep Convolutional GAN (DCGAN) for image generation, as detailed in [51]. The full descriptions of both the generator and discriminator can be found in Tables 2 and 3, respectively.

The generator employs a combination of linear transformations and batch normalization across multiple layers, with ReLU activation functions throughout the hidden layers to introduce non-linearity. The batch normalization layers stabilize and accelerate the training process by normalizing the inputs of each layer, mitigating issues of internal covariate shift [52]. The final layer uses the Tanh activation function. This configuration is designed to progressively reduce the dimensionality of the data while applying transformations to generate new data points.

The discriminator utilizes a linear architecture with Leaky ReLU activation functions for both the input and hidden layers. The final output layer employs a sigmoid activation function. The Leaky ReLU activation helps prevent the dying ReLU problem by allowing a small gradient when the unit is not active.

To evaluate how well the model has learned the probability distribution of the data, various distance metrics between probability distributions can be used. For example, 2-norm distance, Hellinger distance, Kolmogorov-Smirnov statistic, or Wasserstein distance are all possible choices [53]. Here, we use the Mean Squared Error (MSE) distance between the synthetic data generated by the model and the original dataset.

The training parameters used by both the hybrid and classical GANs are described in Table 4. The HQGAN uses Stochastic Gradient Descent (SGD) as the optimizer, with a generator learning rate of 0.3 and a discriminator learning rate of 0.01. It has 30 generator parameters and 2145 discriminator parameters, utilizing a latent vector size of 5. The classical GAN employs the Adam optimizer, with both the generator and discriminator learning rates set to 0.0002, consisting of 805,600 and 183,969 parameters for the generator and discriminator, respectively, with a latent vector size of 100. Both models use Binary Cross-Entropy (BCE) Loss, a batch size of 10, and are trained for 5449 epochs.

All simulations were conducted in Python utilizing the Torch and Pennylane frameworks and executed on the quantum simulator Kuatomu provided by SENAI-CIMATEC. Kuatomu comprises 192 processing cores (CPU) with 384 threads, 3 TB of RAM memory, and 4 NVidia V100 32 GB GPU accelerator cards.

## 4 Results and discussion

In this section, we provide an in-depth overview of the results, followed by an extensive discussion.

In Fig. 4, you can see the learning dynamics of our quantum GAN and the classical GAN. The classical GAN follows a typical pattern, with the Generator loss starting higher than the Discriminator loss and remaining close throughout training. Around the halfway point of training, there is a noticeable change where the Discriminator loss becomes higher than the Generator loss. After this point, both losses stay relatively stable until the end of training. Conversely, for the quantum GAN, the Generator and Discriminator losses initially exhibit

**Table 1** Classical discriminator used for the HQGAN

| Discriminator | | | | |
|---|---|---|---|---|
| Layer | Layer type | In features | Out features | Activation function |
| Input layer | Linear | 16 | 64 | ReLU |
| Hidden layer | Linear | 64 | 16 | ReLU |
| Output layer | Linear | 16 | 1 | Sigmoid |

**Table 2** Detailed architecture of the generator in the classical GAN model, illustrating each layer's type, the number of input and output features, and the activation functions employed

| Generator | | | | |
|---|---|---|---|---|
| Layer | Layer type | In features | Out features | Activation function |
| Input layer | Linear | 100 | 1024 | ReLU |
| Hidden layer | Linear | 1024 | 512 | ReLU |
| Hidden layer | Batch norm 1D | 512 | 512 | ReLU |
| Hidden layer | Linear | 512 | 256 | ReLU |
| Hidden layer | Batch norm 1D | 256 | 256 | ReLU |
| Hidden layer | Linear | 256 | 128 | ReLU |
| Hidden layer | Batch norm 1D | 128 | 128 | ReLU |
| Hidden layer | Linear | 128 | 64 | ReLU |
| Hidden layer | Batch norm 1D | 64 | 64 | ReLU |
| Hidden layer | Linear | 64 | 32 | ReLU |
| Hidden layer | Batch norm 1D | 32 | 32 | ReLU |
| Hidden layer | Linear | 32 | 16 | ReLU |
| Hidden layer | Batch norm 1D | 16 | 16 | ReLU |
| Output layer | Linear | 16 | 16 | Tanh |

The generator uses a combination of linear transformations and batch normalization across multiple layers with ReLU activation functions in the input and hidden layers and a Tanh activation function in the output layer

**Table 3** Detailed architecture of the Discriminator in the classical GAN model, showing each layer type, the number of input and output features, and the activation functions used

| Discriminator | | | | |
|---|---|---|---|---|
| Layer | Layer type | In features | Out features | Activation function |
| Input layer | Linear | 16 | 512 | Leaky ReLU |
| Hidden layer | Linear | 512 | 256 | Leaky ReLU |
| Hidden layer | Linear | 256 | 128 | Leaky ReLU |
| Hidden layer | Linear | 128 | 64 | Leaky ReLU |
| Hidden layer | Linear | 64 | 32 | Leaky ReLU |
| Hidden layer | Linear | 32 | 16 | Leaky ReLU |
| Hidden layer | Linear | 16 | 8 | Leaky ReLU |
| Hidden layer | Linear | 8 | 4 | Leaky ReLU |
| Output layer | Linear | 4 | 1 | Sigmoid |

The discriminator employs a linear structure with Leaky ReLU activation functions across the input and hidden layers, and a sigmoid activation function at the output layer for binary classification

stability, with the Generator loss consistently lower than that of the Discriminator. However, as training progresses, the Generator loss begins to increase, resulting in a divergence between the two loss values.

The comparative analysis of the outcomes produced by both GANs, assessing the MSE distance for the original dataset, is depicted in Fig. 5. In the case of the classical GAN, there is a consistent reduction in the gap between the synthetic and original data, ultimately reaching a stabilized state towards the end of the learning process. In contrast, the quantum GAN exhibits erratic behavior, failing to demonstrate a clear trend indicating convergence towards the real data. Additionally, the MSE value for the quantum GAN is higher than that of the classical counterpart, further highlighting the disparities in performance.

As the quantum GAN exhibited a divergence midway through training, we hypothesized that it might

**Table 4** Comparison of training configurations and parameters for hybrid quantum-classical GAN and classical GAN models

| | HQGAN | Classical GAN |
|---|---|---|
| Optimizer | SGD | Adam |
| Generator learning rate | 0.3 | 0.0002 |
| Discriminator learning rate | 0.01 | 0.0002 |
| Generator parameters | 30 | 805.600 |
| Discriminator parameters | 2145 | 183.969 |
| Latent vector size | 5 | 100 |
| Loss function | BCE loss | |
| Batch size | 10 | |
| Epochs | 5449 | |

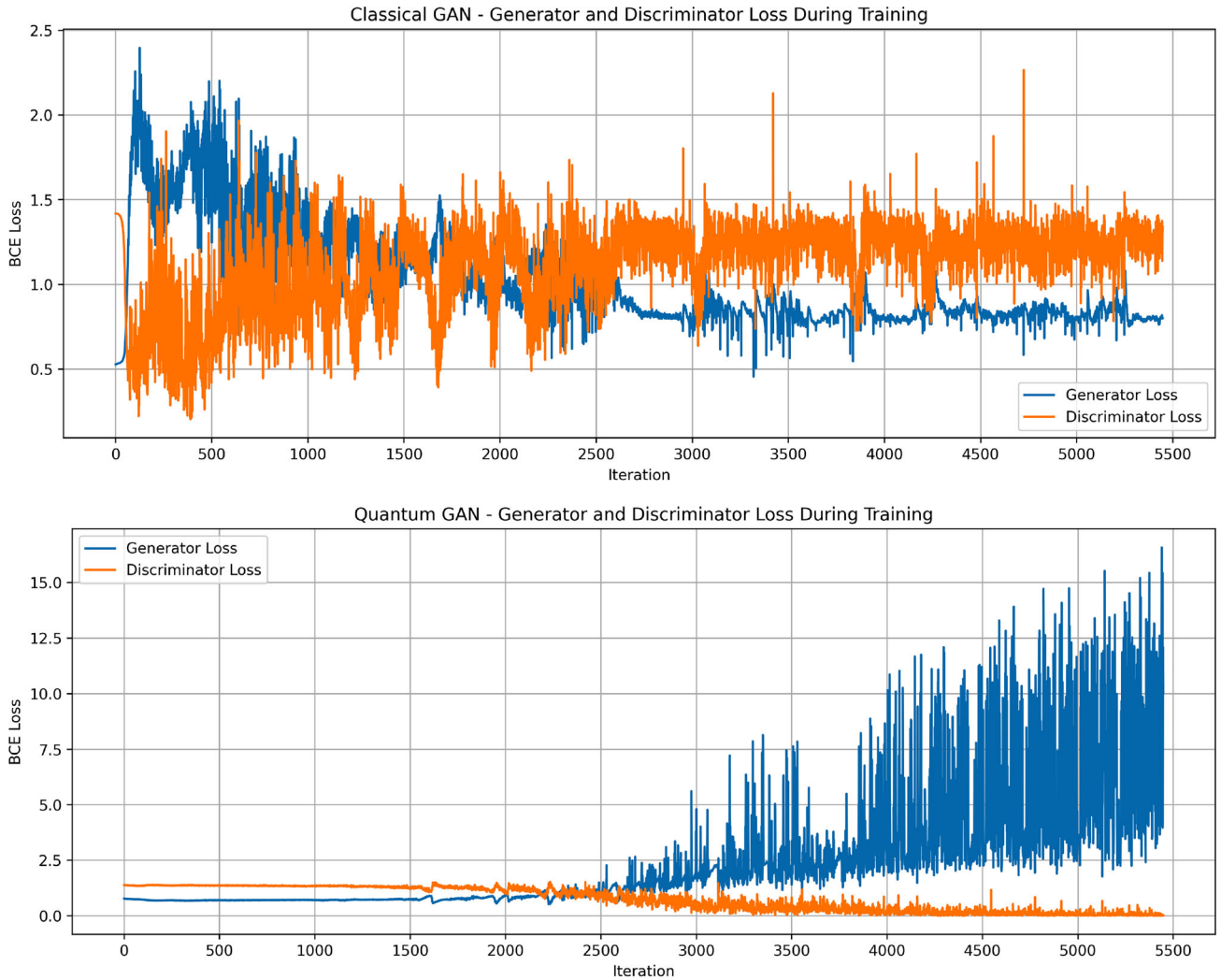


**Fig. 4** Binary cross entropy loss comparison between quantum and classical models: classical GAN shows standard behavior while the quantum GAN shows a divergence around halfway through training

have encountered overfitting, potentially due to the excessive number of training iterations. To validate this assumption, we conducted additional experiments in which we limited the maximum number of iterations to just before the observed divergence, approximately 2500 steps. We applied the same adjustment to the classical GAN in this renewed set of experiments to ensure consistency in the evaluation process.

The new results are depicted in Fig. 6. As anticipated, both quantum and classical GANs exhibited indications

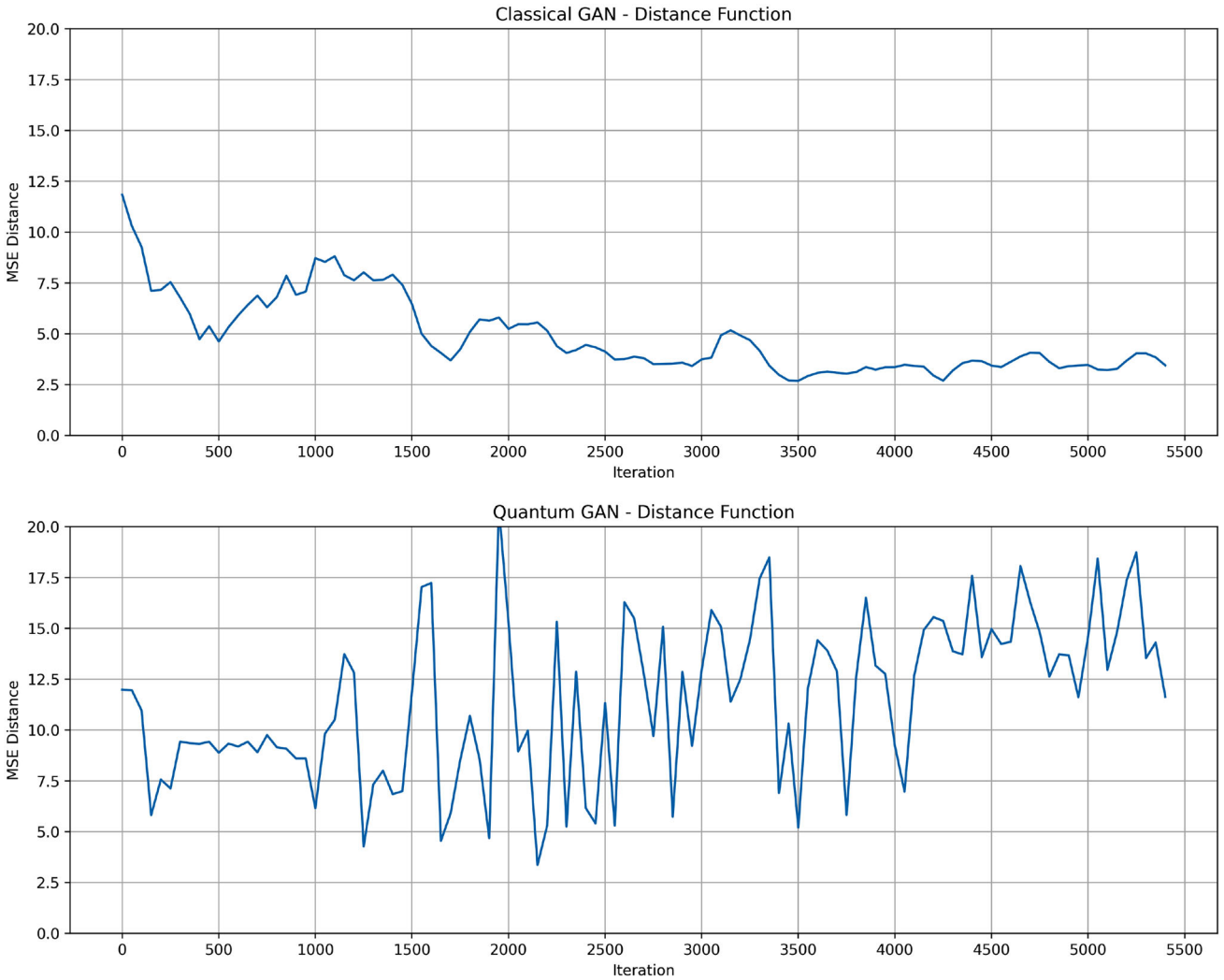


**Fig. 5** Mean square error comparison between synthetic and original data: classical GAN displays decreasing distance between synthetic and original data, stabilizing towards end of training. Conversely, quantum GAN exhibits erratic behavior, failing to approach real data. quantum GAN's Moreover, quantum GAN's MSE is consistently higher than the classical counterpart's

of convergence in their loss graphs. However, when analyzing the MSE distance, the quantum GAN continued to display erratic behavior, with no clear indications of improvement. On the other hand, the classical GAN still demonstrated an improvement in its MSE.

### 4.1 Discussion

The unexpected behavior observed in our hybrid quantum-classical GAN could be influenced by multiple factors warranting detailed investigation. While the quantum component of the GAN may struggle with pattern recognition within this specific dataset regime, we refrain from attributing it as the primary cause and instead consider it as part of a comprehensive analysis. An essential consideration is the possibility of incomplete preprocessing steps applied to the data, suggesting the need for further refinements to enhance learning efficacy. Moreover, optimizing the partial measurement operation on the state $\rho(z)$ used by the generator may be crucial for accurately estimating probabilities of basic states as faithful data representations.

Another critical consideration is the absence of Batch Normalization in the generator of our hybrid quantum-classical GAN, unlike the classical GAN. This architectural difference raises the crucial question: could the lack of Batch Normalization be a decisive factor contributing to the hybrid GAN's struggle to converge to optimal solutions? Batch Normalization is renowned for normalizing activation functions within neural network layers, thereby accelerating convergence, reducing internal covariate shift, and facilitating smoother gradient flow during training. The absence of Batch Normalization in our hybrid model may lead to instability in training dynamics, slower convergence rates, and potentially inferior quality in generated samples.

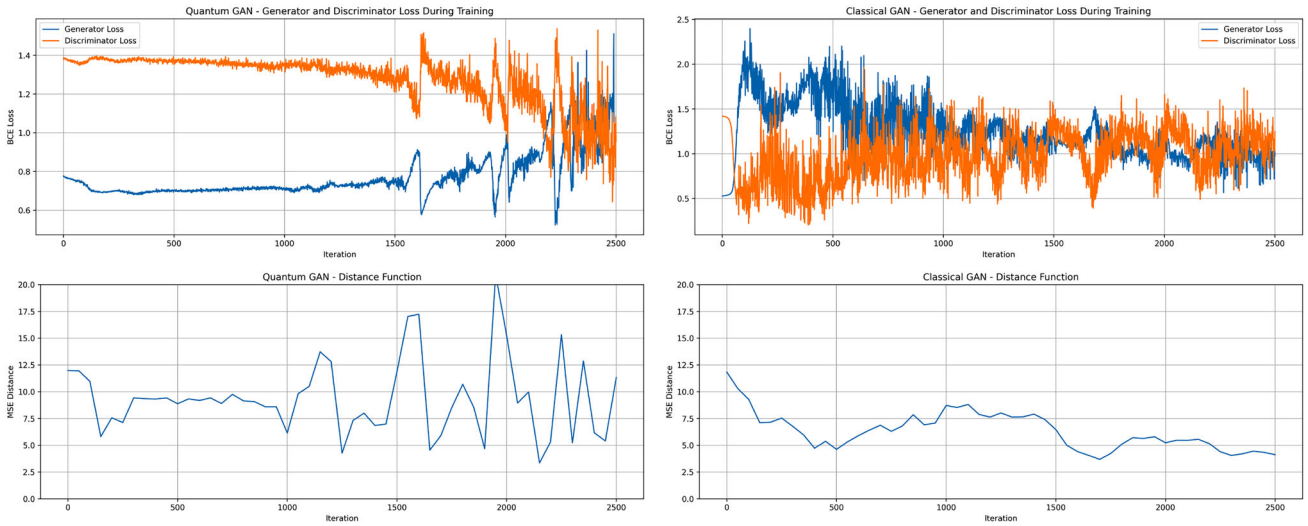


**Fig. 6** Results obtained after truncating training at 2500 iterations. Both quantum and classical GANs show a convergence patterns in their loss graphs. However, while the classical GAN demonstrates improvement in mean squared error (MSE), the quantum GAN still exhibits erratic behavior with no discernible signs of enhancement

The necessity of exploring and understanding the nuanced differences in architecture and training dynamics between classical and quantum models in GAN frameworks is useful in future research endeavors that could explore novel normalization layers specifically for quantum models. Such investigations aim to bolster convergence rates, stabilize training dynamics, and ultimately improve the generation quality of hybrid quantum-classical GANs across a spectrum of generative tasks.

## 5 Conclusions

In conclusion, this study explored the feasibility of using a hybrid quantum-classical generative adversarial network (HQGAN) to generate synthetic financial data reliably, where a quantum circuit serves as the generator and a classical neural network acts as the discriminator, for synthesizing financial data essential for money withdraw approval. Our research article contributes in three key ways: first, it proposes a hybrid quantum-classical generative adversarial network (HQGAN) for synthetic financial data generation. Second, it comprehensively compares HQGAN with a conventional classical GAN, evaluating both models using loss function convergence and MSE distance metrics. Third, we apply these models to real financial data, demonstrating their practical applicability and highlighting their potential in addressing data privacy and integrity challenges in the financial sector.

Our model was applied to a case study utilizing real financial data provided by Banco BV. The dataset encompasses the profiles of bank clients who request to withdraw money from FGTS, a Brazilian governmental fund established to protect workers. The data showed a high imbalance, with only 0.4% of the samples belonging to the minority class. The challenge was to increase the number of samples from the minority class (approved clients) to create a more balanced dataset for training our classification models. We conducted a comparative analysis between our method and a classical GAN to gain insights into the performance of our quantum model. For precise evaluation, we utilized the Mean Squared Error (MSE) as a quantitative measure to gauge the discrepancy between real and generated distributions, alongside analyzing the convergence of loss in both models.

Despite its initial promise, our evaluation showed that HQGAN did not achieve sufficient accuracy to learn the complex patterns present in financial data. This result highlights the current limitations of quantum-inspired approaches to dealing with the complexities of financial datasets. Future research should focus on revisiting our methodology to address the unresolved issues. In addition, we intend to explore alternative research routes, including investigating hybrid GAN architectures with a quantum discriminator instead of a quantum generator or even exploring the potential of fully quantum GANs. These alternative approaches hold promise for possibly capturing the intricate patterns present in financial data.

**Acknowledgement** The authors would like to thank Banco BV for providing resources for the project from which this article is derived, as well as for contributions and discussions throughout the work. Acknowledgements also to the Supercomputing Center for Industrial Innovation (CS2I), the Reference Center for Artificial Intelligence (CRIA), and the Latin American Quantum Computing Center (LAQCC), all from SENAI CIMATEC.

## Author contributions

All authors contributed significantly to the conception and design of the study. Yan Alef C. Silva, Maria Heloísa F. da Silva, and Lucas Q. Galvão conducted the literature review. Otto M. Pires, Mauro Q. Nooblath, and Anton S. Albino developed the methodology. Otto M. Pires and Mauro Q. Nooblath conducted experiments and performed the analysis. The manuscript was collaboratively written by all authors. All authors have read and approved the final manuscript.

**Data Availability Statement** All data used in this research belongs to Banco BV and is not publicly available. The data can be accessed upon permission by Banco BV.



## References

1. N.M. Safdar, J.D. Banja, C.C. Meltzer, Ethical considerations in artificial intelligence. Eur. J. Radiol. **122**, 108768 (2020)
2. D. Zha, Z.P. Bhat, K.-H. Lai, F. Yang, Z. Jiang, S. Zhong, X. Hu, Data-centric artificial intelligence: a survey. arXiv:2303.10158 (2023). https://doi.org/10.48550/arXiv.2303.10158
3. M. Kim, H. Ahn, Quality control and verification of artificial intelligence data. GEO DATA (2021). https://doi.org/10.22761/dj2021.3.3.004
4. C. Kuner, F.H. Cate, O. Lynskey, C. Millard, N. Ni Loideain, D.J.B. Svantesson, *Expanding the Artificial Intelligence-data Protection Debate* (Oxford University Press, Oxford, 2018)
5. M. Awasthi, Verifiable and practical compliance for data privacy laws, in 2022 IEEE 29th International Conference on High Performance Computing, Data and Analytics Workshop (HiPCW), pp. 59–60 (2022) https://doi.org/10.1109/HiPCW57629.2022.00013
6. F. Kreuter, G. Haas, F. Keusch, S. Bähr, M. Trappmann, Collecting survey and smartphone sensor data with an app: opportunities and challenges around privacy and informed consent. Social Sci. Comput. Rev. **38**, 533–549 (2020). https://doi.org/10.1177/0894439318816389
7. S. James, C. Harbron, J. Branson, M. Sundler, Synthetic data use: exploring use cases to optimise data utility. Discover Artif. Intell. **1**(1), 15 (2021)
8. D.J. McDuff, T.R. Curran, A. Kadambi, Synthetic data in healthcare. arXiv:2304.03243 (2023). https://doi.org/10.48550/arXiv.2304.03243
9. K. Man, J. Chahl, A review of synthetic image data and its use in computer vision. J. Imaging (2022). https://doi.org/10.3390/jimaging8110310
10. I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, Generative adversarial networks. Commun. ACM **63**(11), 139–144 (2020). https://doi.org/10.1145/3422622
11. T.A. Ngo, T. Nguyen, T.C. Thang, A survey of recent advances in quantum generative adversarial networks. Electronics **12**(4), 856 (2023)
12. A. Dash, J. Ye, G. Wang, A review of generative adversarial networks (GANS) and its applications in a wide variety of disciplines: from medical to remote sensing. IEEE Access **12**, 18330–18357 (2024). https://doi.org/10.1109/ACCESS.2023.3346273
13. W.M. Kouw, M. Loog, An introduction to domain adaptation and transfer learning. Preprint arXiv:1812.11806 (2018)
14. S. Takahashi, Y. Chen, K. Tanaka-Ishii, Modeling financial time-series with generative adversarial networks. Phys. A **527**, 121261 (2019)
15. D. Efimov, D. Xu, L. Kong, A. Nefedov, A. Anandakrishnan, Using generative adversarial networks to synthesize artificial financial datasets. Preprint arXiv:2002.02271 (2020)
16. X. Zhou, Z. Pan, G. Hu, S. Tang, C. Zhao, Stock market prediction on high-frequency data using generative adversarial nets. Math. Problems Eng. **2018**, 1 (2018)
17. H. Lin, C. Chen, G. Huang, A. Jafari, Stock price prediction using generative adversarial networks. J. Comput. Sci. **17**(3), 188–196 (2021). https://doi.org/10.3844/jcssp.2021.188.196
18. C. Jiang, W. Lu, Z. Wang, Y. Ding, Benchmarking state-of-the-art imbalanced data learning approaches for credit scoring. Expert Syst. Appl. **213**, 118878 (2023). https://doi.org/10.1016/j.eswa.2022.118878
19. T. Leangarun, P. Tangamchit, S. Thajchayapong, Stock price manipulation detection using generative adversarial networks. In: 2018 IEEE Symposium Series on Computational Intelligence (SSCI), pp. 2104–2111. IEEE (2018)
20. S.Z. Aftabi, A. Ahmadi, S. Farzi, Fraud detection in financial statements using data mining and GAN models. Expert Syst. Appl. **227**, 120144 (2023). https://doi.org/10.1016/j.eswa.2023.120144
21. X. Wang, K. Yu, S. Wu, J. Gu, Y. Liu, C. Dong, Y. Qiao, C. Change Loy, Esrgan: enhanced super-resolution generative adversarial networks. In: Proceedings of the European Conference on Computer Vision (ECCV) Workshops (2018)
22. J. Yu, Z. Lin, J. Yang, X. Shen, X. Lu, T.S. Huang, Generative image inpainting with contextual attention. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 5505–5514 (2018)
23. T. Li, S. Zhang, J. Xia, Quantum generative adversarial network: a survey. Comput. Mater. Cont. **64**(1), 401–438 (2020)

24. D. Repecka, V. Jauniskis, L. Karpus, E. Rembeza, I. Rokaitis, J. Zrimec, S. Poviloniene, A. Laurynenas, S. Viknander, W. Abuajwa et al., Expanding functional protein sequence spaces using generative adversarial networks. Nat. Mach. Intell. **3**(4), 324–333 (2021)
25. Z. Zhang, F. Li, J. Guan, Z. Kong, L. Shi, S. Zhou, Gans for molecule generation in drug design and discovery. In: Generative Adversarial Learning: Architectures and Applications, pp. 233–273. Springer (2022)
26. J. Li, R.O. Topaloglu, S. Ghosh, Quantum generative models for small molecule drug discovery. IEEE Trans. Quantum Eng. **2**, 1–8 (2021)
27. J. Engel, K.K. Agrawal, S. Chen, I. Gulrajani, C. Donahue, A. Roberts, Gansynth: adversarial neural audio synthesis. Preprint arXiv:1902.08710 (2019)
28. Z. Hu, J. Wang, A novel adversarial inference framework for video prediction with action control. In: Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops (2019)
29. M. Uricár, P. Krizek, D. Hurych, I. Sobh, S. Yogamani, P. Denny, Yes, we GAN: applying adversarial techniques for autonomous driving. Preprint arXiv:1902.03442 (2019)
30. C. Besombes, O. Pannekoucke, C. Lapeyre, B. Sanderson, O. Thual, Producing realistic climate data with generative adversarial networks. Nonlinear Process. Geophys. **28**(3), 347–370 (2021)
31. M.J. Smith, J.E. Geach, Generative deep fields: arbitrarily sized, random synthetic astronomical images through deep learning. Mon. Not. R. Astron. Soc. **490**(4), 4985–4990 (2019)
32. I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, Generative adversarial nets. Adv. Neural Inform. Process. Syst. **27**, 1 (2014)
33. P.-L. Dallaire-Demers, N. Killoran, Quantum generative adversarial networks. Phys. Rev. A **98**, 012324 (2018). https://doi.org/10.1103/PhysRevA.98.012324
34. S. Lloyd, C. Weedbrook, Quantum generative adversarial learning. Phys. Rev. Lett. **121**, 040502 (2018). https://doi.org/10.1103/PhysRevLett.121.040502
35. X. Gao, Z.-Y. Zhang, L.-M. Duan, A quantum machine learning algorithm based on generative models. Sci. Adv. **4**(12), 9004 (2018)
36. H.-L. Huang, Y. Du, M. Gong, Y. Zhao, Y. Wu, C. Wang, S. Li, F. Liang, J. Lin, Y. Xu, R. Yang, T. Liu, M.-H. Hsieh, H. Deng, H. Rong, C.-Z. Peng, C.-Y. Lu, Y.-A. Chen, D. Tao, X. Zhu, J.-W. Pan, Experimental quantum generative adversarial networks for image generation. Phys. Rev. Appl. **16**, 024051 (2021). https://doi.org/10.1103/PhysRevApplied.16.024051
37. A. Creswell, T. White, V. Dumoulin, K. Arulkumaran, B. Sengupta, A.A. Bharath, Generative adversarial networks: an overview. IEEE Signal Process. Mag. **35**(1), 53–65 (2018). https://doi.org/10.1109/MSP.2017.2765202
38. J.E. Pennylane, Quantum GANs. https://pennylane.ai/qml/demos/tutorial_quantum_gans/. Accessed on 15 February 2024 (2023)
39. P. Bojanowski, A. Joulin, D. Lopez-Paz, A. Szlam, Optimizing the latent space of generative networks. Preprint arXiv:1707.05776 (2017)
40. Y. Hong, U. Hwang, J. Yoo, S. Yoon, How generative adversarial networks and their variants work: an overview. ACM Comput. Surv. (CSUR) **52**(1), 1–43 (2019)
41. A. Radford, L. Metz, S. Chintala, Unsupervised representation learning with deep convolutional generative adversarial networks. Preprint arXiv:1511.06434 (2015)
42. M. Mirza, S. Osindero, Conditional generative adversarial nets. Preprint arXiv:1411.1784 (2014)
43. J. Donahue, P. Krähenbühl, T. Darrell, Adversarial feature learning. Preprint arXiv:1605.09782 (2016)
44. S. Hitawala, Comparative study on generative adversarial networks. Preprint arXiv:1801.04271 (2018)
45. W. Huggins, P. Patil, B. Mitchell, K.B. Whaley, E.M. Stoudenmire, Towards quantum machine learning with tensor networks. Quantum Sci. Technol. **4**(2), 024001 (2019)
46. J. Preskill, Quantum computing in the NISQ era and beyond. Quantum **2**, 79 (2018)
47. Visão Geral-Sobre o FGTS. https://www.fgts.gov.br/Pages/sobre-fgts/visao-geral.aspx. Accessed on 06 June 2024 (2024)
48. Datasets: Imbalanced datasets. https://developers.google.com/machine-learning/crash-course/overfitting/imbalanced-datasets. Accessed on 23 August 2024 (2024)
49. E. Combarro, S. Gonzalez-Castillo, *A Practical Guide to Quantum Machine Learning and Quantum Optimization: Hands—On Approach to Modern Quantum Algorithms*, 1st edn. (Packt Publishing, London, 2023)
50. Pytorch Documentation—Linear. https://pytorch.org/docs/stable/generated/torch.nn.Linear.html. Accessed on 09 June 2024 (2024)
51. N.I. PyTorch, DCGAN Tutorial. https://pytorch.org/tutorials/beginner/dcgan_faces_tutorial.html. Accessed on 07 June 2024 (2024)
52. S. Ioffe, C. Szegedy, Batch normalization: accelerating deep network training by reducing internal covariate shift (2015)
53. C.A. Riofr'io, O. Mitevski, C. Jones, F. Krellner, A. Vuvckovi'c, J. Doetsch, J. Klepsch, T. Ehmer, A. Luckow, A performance characterization of quantum generative models. (2023). https://api.semanticscholar.org/CorpusID:256105777