particles

MDPI

Article

Long-Lived Particles Anomaly Detection with Parametrized Quantum Circuits

Simone Bordoni [1,†], Denis Stanev [2,†], Tommaso Santantonio [1] and Stefano Giagu [1,*]

[1] Physics Department, Sapienza Università di Roma, 00185 Rome, Italy
[2] Gran Sasso Science Institute (GSSI), 67100 L'Aquila, Italy
[*] Correspondence: stefano.giagu@uniroma1.it; Tel.: +39-0649914407
[†] These authors contributed equally to this work.

Abstract: We investigate the possibility to apply quantum machine learning techniques for data analysis, with particular regard to an interesting use-case in high-energy physics. We propose an anomaly detection algorithm based on a parametrized quantum circuit. This algorithm was trained on a classical computer and tested with simulations as well as on real quantum hardware. Tests on NISQ devices were performed with IBM quantum computers. For the execution on quantum hardware, specific hardware-driven adaptations were devised and implemented. The quantum anomaly detection algorithm was able to detect simple anomalies such as different characters in handwritten digits as well as more complex structures such as anomalous patterns in the particle detectors produced by the decay products of long-lived particles produced at a collider experiment. For the high-energy physics application, the performance was estimated in simulation only, as the quantum circuit was not simple enough to be executed on the available quantum hardware platform. This work demonstrates that it is possible to perform anomaly detection with quantum algorithms; however, as an amplitude encoding of classical data is required for the task, due to the noise level in the available quantum hardware platform, the current implementation cannot outperform classic anomaly detection algorithms based on deep neural networks.

Keywords: particle physics; long-lived particles; anomaly detection; quantum computing; machine learning; quantum machine learning; parametrized quantum circuits

check for updates

Citation: Bordoni, S.; Stanev, D.; Santantonio, T.; Giagu, S. Long-Lived Particles Anomaly Detection with Parametrized Quantum Circuits. *Particles* **2023**, *6*, 297–311. https://doi.org/10.3390/particles6010016

Academic Editors: Armen Sedrakian and Omar Benhar

Received: 14 December 2022
Revised: 7 February 2023
Accepted: 9 February 2023
Published: 13 February 2023

1. Introduction

With the contemporary peak in interest regarding machine learning algorithms for their many applications in scientific research, we are also witnessing a rapid acceleration in the concurrent development of quantum computing. A combination of these two research fields has led to the development of quantum machine learning (QML) algorithms [1–4].

In this work, we propose a quantum version of a classic machine learning algorithm known as anomaly detection. This algorithm is implemented with an artificial neural network, in particular an autoencoder architecture [5–7]. In quantum machine learning, the autoencoder is realized using parametrized quantum circuits [8–10].

First, we test a simpler version of the quantum algorithm in an easier task involving a standard benchmark dataset in machine learning, the handwritten digits MNIST dataset. We then apply the technique to a more complex and interesting use-case, the identification of anomalous signatures inside a particle detector due to the decay of long-lived particles with macroscopic lifetimes. The application of quantum machine learning to high-energy physics is an interesting field that has been studied using QML simulators in some recent works [11–21]. In this paper, we present the first application of QML to the task of anomaly detection for long-lived particle identification and also prove that the proposed variational quantum circuits could be used on actual quantum hardware. The parametrized quantum circuits developed in this work are in fact simple enough to be tested on noisy intermediate-scale quantum (NISQ) computers [22–25]. The tests on real quantum hardware have been implemented on IBM quantum computers [26]. We stress here that the proposed algorithm is not meant, at this stage, to outperform its classical counterpart on classical data, but only to show that it is possible to use quantum variational algorithms for anomaly detection. QML algorithms will be employed in real applications only when less noisy qubits or effective error correction procedures are available in digital quantum computers. The proposed algorithm may show future advantages for the analysis of quantum data [27]. In fact, it is really difficult to manage quantum data with classical hardware, due to the information size growing exponentially with respect to the number of qubits. On the other hand, a QML algorithm can directly analyze quantum data, overcoming the problem of amplitude encoding (Section 4.1).

The paper is structured in the following way: In Section 2, the fundamental concepts underlying anomaly detection algorithms and parametrized quantum circuits are briefly reviewed. Section 3 describes the strategy used to develop the quantum algorithms and the performance estimated using a simulator of the quantum circuits on classic hardware. In Section 4, we describe the changes we have implemented in the quantum circuit to be executed on the IBM_hanoi (IBM_hanoi: https://quantum-computing.ibm.com/services/resources?tab=systems&skip=10&system=ibm_hanoi (accessed on 10 February 2023)) quantum computer and discuss the results. Conclusions and future developments are presented in Section 5.

2. Background

In this section we briefly recall the main elements needed to understand the proposed quantum anomaly detection algorithm.

2.1. Anomaly Detection Algorithms

Anomaly detection describes a class of algorithms that aims at the identification of rare items, events or observations, which deviate significantly from the majority of the data and do not conform to a well-defined notion of normal behavior. Anomaly detection has recently gained increasing interest in experiments at the Large Hadron Collider (LHC) [28], as a viable machine learning approach to implement signal model-independent searches for new physics effects. The technique in fact requires only a precise prediction of the background (normal data) to train a classifier model to distinguish data from the background, without requiring a specific description of the new physics signal (anomalous data) [21].

One way to implement an anomaly detection algorithm is to train a particular artificial neural network (ANN) architecture called autoencoder [5–7], utilized in various applications of unsupervised learning. An autoencoder is composed of two main parts: an encoder and a decoder (Figure 1). The encoder compresses the initial data down to a small dimension (called latent dimension). The decoder inverts the process to reconstruct the original data from the compressed representation. The parameters of the neural network are trained in order to minimize the difference between the initial and reconstructed data. The loss function (also called reconstruction loss) is therefore a measure of how accurately the reconstructed data resemble the original. For anomaly detection, the autoencoder is trained only on data samples belonging to the normal event class (e.g., background). When the trained model is applied to new samples, we expect the loss function to have different values for normal and anomalous data. By choosing a threshold value for the loss function it is possible to classify an input based on whether its reconstruction loss lands above or below this threshold. The performance of the trained algorithm is usually presented in terms of the receiver operating characteristic (ROC) curve, which shows the true positive rate as a function of the false positive rate at different classification thresholds [29] and in terms of the area under the ROC curve (AUC), which provides an aggregate measure of performance across all possible classification thresholds.

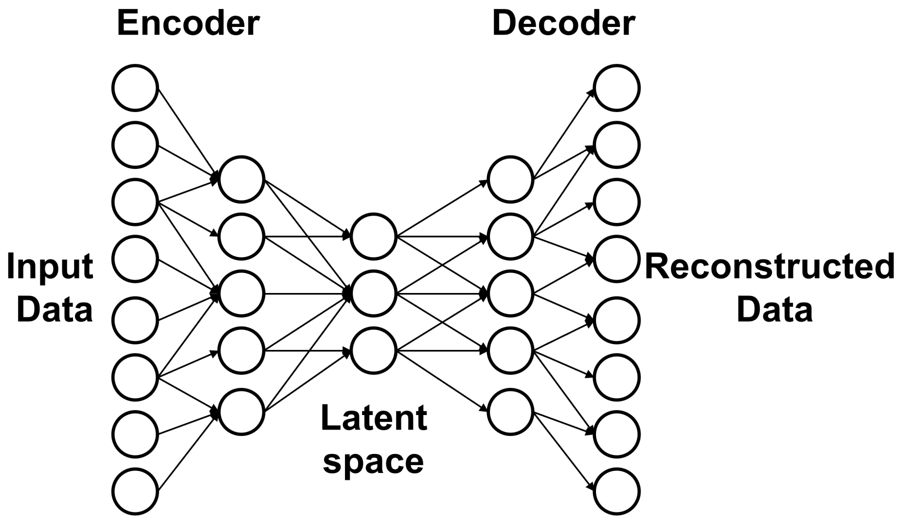


Figure 1. Schematic representation of an autoencoder. The input data are compressed by the encoder to a smaller number of features in the latent space. The decoder tries to reconstruct the original data from the compressed one.

2.2. Parametrized Quantum Circuits

A parametrized quantum circuit (PQC) is a quantum algorithm that depends on free parameters and that can be used as the quantum counterpart of classical ANNs. In this kind of circuits, the input information is stored in the initial state of the qubits. It can be stored as the phase (phase encoding) or in the states amplitudes (amplitude encoding) [30–32]. The initial state is transformed using rotation gates and entangling gates, usually controlled-not (C-NOT) gates [33]. These gates can be organized in layers; in our circuit architecture, one layer was composed of rotation gates (R_x, R_y, R_z) acting on all qubits followed by a series of C-NOT gates coupling neighboring qubits (Figure 2). The trainable weights are the angles of rotation gates and can be trained using the conventional stochastic gradient descent techniques via backpropagation adopted in the training of ANNs [34].

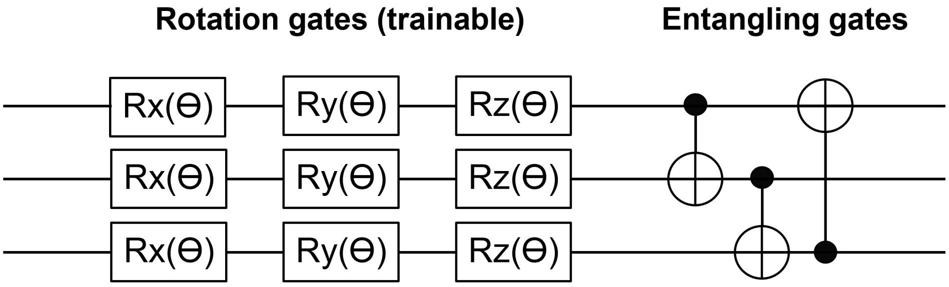


Figure 2. Circuit representation for three qubits of one layer of the PQC used in this work. Single qubit rotation gates with trainable rotation angles are followed by an entangling layer made of C-NOT gates acting on neighboring qubits.

A quantum circuit implements a unitary, thus invertible, transformation on the initial state. This represents a great advantage for the autoencoder architecture, as the decoder can be taken as simply the inverse of the encoder quantum circuit (Figure 3). In order to compress information, the encoder circuit has to disentangle and set to zero state a given number of qubits [35]. The loss function is thus taken as the expected measurement values of these qubits. In this way, for the training of the circuit, only the encoder.is necessary.

For the simulation and training of the PQC, we used the QIBO [36] library that can be easily integrated with Tensorflow [37] for automatic differentiation.

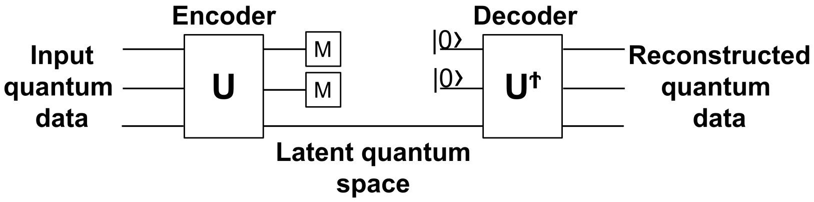


Figure 3. Schematic representation of a quantum autoencoder. The encoder acts as a unitary transformation U that tries to disentangle a certain number of qubits (force them to the $|0\rangle$ state). In this way, the initial quantum information is compressed into a latent quantum space. The decoder implements the inverse transformation $U^{\dagger}$ to reconstruct the original data. The loss function is taken as the measurement expectation value of the qubits that are eliminated after the compression.

3. Simulation on Classic Hardware

In order to find the best parameters of the proposed anomaly detection algorithm, the first tests were carried out with simulations on classical hardware. The anomaly detection task can be solved, with satisfactory discriminative power, by requiring a PQC with enough expressive power. This can be achieved by increasing the depth of the circuit. However, increasing the depth of the circuit may lead to difficulties in finding a suitable minimum in the loss function during the training, mainly due to the presence of barren plateaus [38,39]. To reach an acceptable trade-off between the two effects, a detailed study of the different possible quantum gate topologies that could be used in the circuit was carried out. The algorithm was applied at first for the recognition of anomalous handwritten digits (Section 3.1) and then for the high-energy physics problem (Section 3.2). In both cases, the quantum encoding of the classical input data was implemented using amplitude encoding. In this way, it was possible to encode a number of features that grew exponentially with the number of qubits in the quantum circuit. A drawback with the amplitude encoding, on the other hand, was in the state preparation that required an exponential number of gates [40] with respect to the number of qubits. The state preparation is a current and still open problem on NISQ devices.

3.1. Simple Use-Case: Handwritten Digits

The anomaly detection on handwritten digits was carried out on the MNIST dataset. We defined handwritten “zero” digits as normal data and “one” digits as anomalous data. Examples of two images from the MNIST dataset are shown in Figure 4.

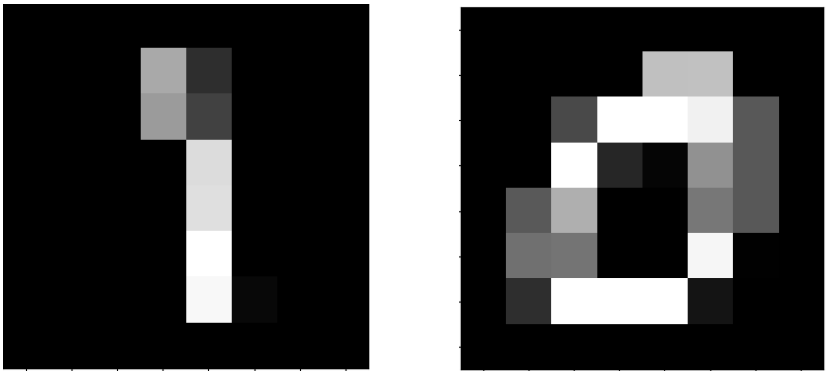


Figure 4. Example of images for the “zero” and “one” digits for the MNIST handwritten digits dataset. Images were compressed to 8 × 8 pixels.

The original MNIST images were compressed down to 8 × 8 pixels; every pixel was an integer number from 0 to 255 (8-bit grayscale). In order to encode the classical data, the images were flattened to obtain feature vectors composed of 64 elements. These vectors were then normalized so that they could be encoded in the state amplitudes of six qubits using amplitude encoding (Section 2.2).

We tested the PQC with a different number of layers and number of compressed qubits. The number of layers was varied from four to eight. This interval was chosen because circuits with less than four layers did not have enough expressive power to solve the task. On the other hand, incrementing the number of layers to more than eight made the training procedure complicated because of barren plateaus. The number of compressed qubits was varied from one to four. In this range, the decoder was able to reconstruct the original images with a good precision.

The performance of the different circuit architectures was compared using the AUC value (Section 2.1). The best configuration was found with six layers and three compressed qubits. This procedure was repeated in order to find the best entangling gates ansatz. We tested different C-NOT configurations, and the one that produced the best performance is reported in Figure 5. In order to improve the performance, rotation gates with trainable parameters were added at the end of the encoder circuit for the three compressed qubits. A summary of the employed circuit is reported in Figure 5. It is worth noticing here that the chosen configuration required only nearest neighbor connectivity for six qubits placed in a ring topology. We also verified that was possible to preserve a good performance by reducing the number of layers from six to four. We leveraged these two properties during the implementation on quantum hardware of the PQC (Section 4.1), to minimize the disruption of the algorithm performance due to the noisy device.

```
q0:     ─RX─RY─RZ─o─────────X─RX─RY─RZ─o─────────X─RX─RY─RZ─o─────────X─RX─RY─ ...
q1:     ─RX─RY─RZ─X─o───────|─RX─RY─RZ─X─o───────|─RX─RY─RZ─X─o───────|─RX─RY─ ...
q2:     ─RX─RY─RZ───X─o─────|─RX─RY─RZ───X─o─────|─RX─RY─RZ───X─o─────|─RX─RY─ ...
q3:     ─RX─RY─RZ─────X─o───|─RX─RY─RZ─────X─o───|─RX─RY─RZ─────X─o───|─RX─RY─ ...
q4:     ─RX─RY─RZ───────X─o─|─RX─RY─RZ───────X─o─|─RX─RY─RZ───────X─o─|─RX─RY─ ...
q5:     ─RX─RY─RZ─────────X─o─RX─RY─RZ─────────X─o─RX─RY─RZ─────────X─o─RX─RY─ ...

q0: ... RZ─o─────────X─RX─RY─RZ─o─────────X─RX─RY─RZ─o─────────X─RX─RY─RZ─
q1: ... RZ─X─o───────|─RX─RY─RZ─X─o───────|─RX─RY─RZ─X─o───────|─RX─RY─RZ─
q2: ... RZ───X─o─────|─RX─RY─RZ───X─o─────|─RX─RY─RZ───X─o─────|─RX─RY─RZ─
q3: ... RZ─────X─o───|─RX─RY─RZ─────X─o───|─RX─RY─RZ─────X─o───|──────────
q4: ... RZ───────X─o─|─RX─RY─RZ───────X─o─|─RX─RY─RZ───────X─o─|──────────
q5: ... RZ─────────X─o─RX─RY─RZ─────────X─o─RX─RY─RZ─────────X─o──────────
```

Figure 5. Circuit representation of the quantum encoder used for anomaly detection of handwritten digits. The encoder circuit is composed of six qubits and six layers. At the end of the quantum circuit, the first three qubits are measured in order to compress information on the three remaining qubits. The loss function is computed as the sum of the probabilities of having any of the measured qubits in the ground state.

For the training of the circuit, a dataset of 5000 images of zero handwritten digits was employed. As the first three qubits were forced to the $|1\rangle$ state in order to apply the data compression, the loss function was the sum of the probabilities of having any of these three qubits in the ground state. Training was performed for 20 epochs using the Adam optimizer [41], with a dynamic learning rate that spanned from 0.4 in the first epochs to 0.001 in the last ones. This variable learning rate helped reduce the problem of barren plateaus. The number of epochs was sufficient to reach a plateau in the loss function. No overfitting was observed during the training process; thus, no early stopping was required. For the training, we used batches composed of 20 samples (250 steps per epoch).

To test the anomaly detection algorithm after the training phase, we used 2000 normal data images not used in the training and 2000 anomalous data images. Figure 6 shows the loss distribution for the two test datasets. It is possible to observe a clear separation, with an average loss value for normal data of 0.307 and 1.026 for anomalous data, with an average root mean square (RMS) of the two distributions of 0.110 and 0.066, respectively.

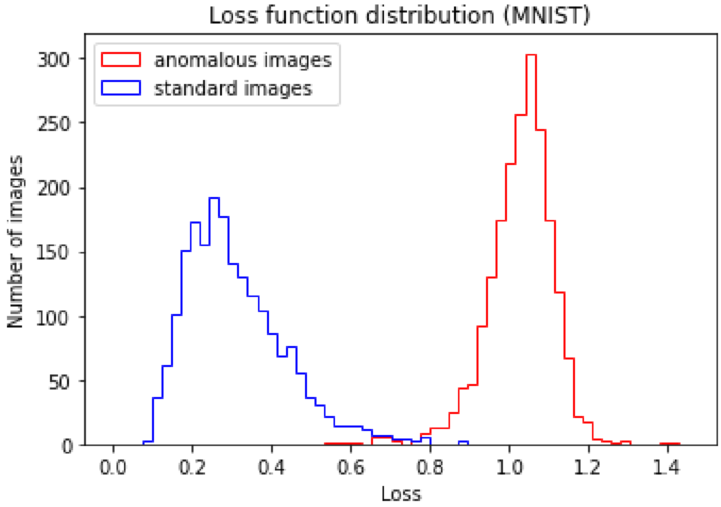


Figure 6. Quantum autoencoder loss function values distribution. The graph was made using 2000 normal data images (zeros) and 2000 anomalous data images (ones) from the MNIST dataset.

3.2. High-Energy Physics Use-Case

An interesting use-case for a high-energy physics application of an anomaly detection algorithm is the identification of anomalous patterns in the triggers system of a collider experiment. As a benchmark scenario for this work, we investigated the specific case of long-lived exotic particles predicted in new physics models with a secluded sector, as in [42,43]. New particles predicted by theories beyond the Standard Model can generically have lifetimes that are long compared to Standard Model particles at the weak scale. When produced at the LHC, such long-lived particles can decay far from the primary interaction vertex and possibly themselves interact with the detector material, leading to a plethora of possible detector signatures. Such signatures are distinctly different from those associated with traditional searches for prompt particles and requires dedicated reconstruction and identification algorithms, especially in the trigger systems of the collider detectors. As an example case, it is possible to think about the high-level muon trigger system of the ATLAS experiment [44], where the computational demands and the lack of flexibility of traditional secondary vertex reconstruction algorithms in the muon spectrometer [45] restrict their applicability only to the offline reconstruction [46]. This prevents their use in the trigger system of the experiment, limiting the discovery potential of the experiment for this kind of new physics searches. The ATLAS experiment muon system aims to collect all the particle hit information from different subdetectors to find muon candidates in a given sector (i.e., a solid angle region of the detector). In this proof of concept study, we restricted our attention only to the barrel region of the muon spectrometer and to the muon drift tube precision detector (MDT). The trigger algorithm tried to find patterns of hits consistent with the presence of muons originating from the same production vertex. One could think to arrange the MDT detector hits into image-like objects, to be used as input for ML algorithms particularly suitable to find patterns such as the muon tracks in this test scenario. Using the published ATLAS detector geometry and resolution as well as its magnetic field map [44], it was possible to generate toy events with muon tracks, from the decays of a noninteracting, neutral, long-lived particle at different decay lengths from the primary proton–proton interaction vertex.

The images were produced initially with a pixel resolution that roughly corresponded to the MDT detector segmentation, each horizontal pixel corresponding to the center of an MDT and each vertical pixel corresponding to one of the layers of the MDT in the sector. The simplified simulation was only parametric, associated a binary value to each pixel (zero or one) depending if one of the muons from the long-lived particle decay went through the tube, and then applied a position smearing based on the published ATLAS detector performance.

A random hit background emulating the typical noise rate conditions during the LHC runs was also included in the simulation. The background noise was generated only accounting for the average hit rate in the spectrometer MDTs, therefore it did not consider correlated backgrounds. We did not aim here to perfectly reproduce the experimental conditions but to give a proof of principle of the anomaly detection algorithm in the context of a high-energy physics experiment.

The images obtained in the simplified simulation were too large (20 × 333 pixels) to be encoded as quantum input in the available quantum computers, therefore they were compressed with a pooling operation to reduce their size to 20 × 100 pixels. The pixels of the compressed image contained the integral of the original pixel content pooled and therefore the image appeared as a grayscale and no longer as a binary image.

Events were generated with a single particle gun generator producing the decay products of a hypothetical dark particle promptly decaying in multiple muons in the primary vertex of the detector and then translated at different decay lengths. The momentum distribution of the particle mimicked the typical expected one for a dark photon in the Falkowski–Ruderman–Volansky–Zupan model [42]; the mass of the long-lived particle was randomly chosen in the range [0.5, 5] GeV.

Two datasets were generated, one corresponding to prompt short-decay-length decays in multimuons (from two to ten muons), with radial decay lengths uniformly distributed between 0.0 and 20.0 cm from the primary interaction vertex, and one corresponding to very displaced decays in multimuons, with radial decay lengths uniformly distributed between 250.0 and 450.0 cm.

Data were conveniently represented in the form of images of dimension 100 × 20 pixels (Figure 7). The encoding procedure used for this dataset was the same as the one described for the MNIST dataset (Section 3.1). In this case, 11 qubits were required for the amplitude encoding.

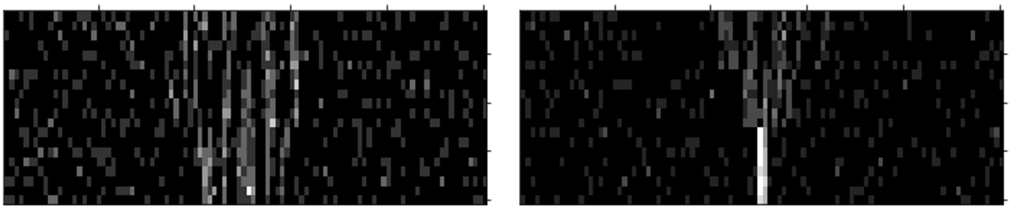


Figure 7. A normal (**left**) and an anomalous (**right**) event used for quantum anomaly detection applied to long-lived particles detection. Normal and anomalous data are, respectively, prompt and displaced decays in multimuons (from two to ten muons). Hits produced by the muons were reconstructed in a toy simulation of the ATLAS MDT chambers that included a random hit background noise mimicking the expected ATLAS phase-2 noise. Data are represented in the form of 100 × 20 pixels images.

In order to find the best PQC ansatz, the same procedure described in Section 3.1 was repeated. For this task, we tested PQCs with a number of layers varying from 6 to 10 and a number of compressed qubits varying from one to four. Choosing a number of layers between this interval represented a good trade-off between the expressive power and the problem of barren plateaus. The best performance (AUC value) was obtained with eight layers and a three-qubit compression.

Training was performed on a dataset of 8000 normal data images for 60 epochs. The number of epochs was sufficient to reach a plateau in the loss function. In this case as well, no overfitting was observed during the training process, thus no early stopping was required. For the training, we used batches composed of 20 samples (400 steps per epoch). The optimizer and learning rate were the same as those used in Section 3.1. In order to mitigate barren plateaus, we also tried to train the circuit layer by layer, as suggested in [47], but the final result was worse than a single all-layers training.

In order to benchmark the performance of the quantum algorithm, we made a comparison with a classic anomaly detection algorithm. The standard autoencoder was implemented with a convolutional neural network (CNN). The encoder CNN was composed of three convolution layers followed by a dense layer and compressed the input features down to a latent space of dimension five. The decoder CNN was composed of a dense layer followed by three transpose convolution layers that restored the original input dimension. The total number of parameters of this neural network was about 7.9×10^3. The loss function was the binary cross-entropy between the input and the reconstructed image. Training was performed on the same dataset used in the quantum case, for 60 epochs and using the Adam optimizer.

The final performance evaluation was carried out on 2000 normal data images and 2000 images of anomalous decays for both the quantum and classic algorithms. The loss function distributions for the normal and anomalous datasets are reported in Figure 8. It is possible to observe a separation in the loss function distribution for anomalous and normal data images in both cases. For the quantum algorithm, the resulting average loss for normal events was 0.870 while the average loss for anomalous events was 0.788. The RMS of the two distributions were, respectively, 0.038 and 0.030. For the classic algorithm, the resulting average loss for normal events was 0.409 while the average loss for anomalous events was 0.355. The RMS of both distributions was 0.017. Note that as the loss functions were different for the classic and quantum case only the relative separations could be compared. It is worth noticing here that, for both algorithms, the anomalous data showed a lower average loss than normal data. This was due to the fact that anomalous images had a simpler structure with hits distributed in a narrow cone with respect to the normal images and thus were easier to compress.

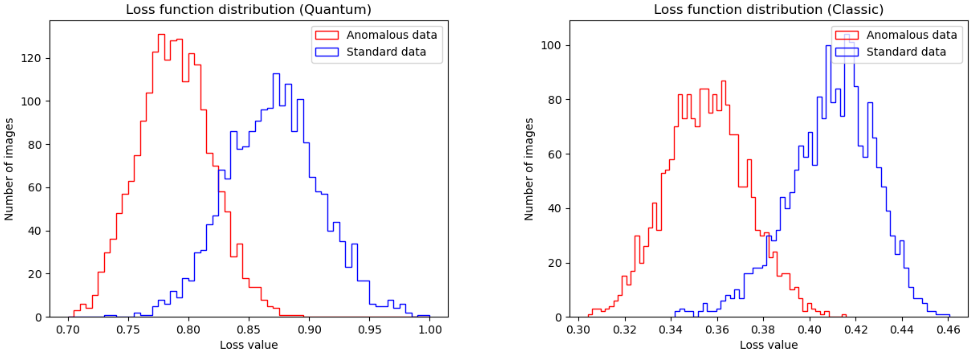


Figure 8. Loss function values distribution for the quantum anomaly detection algorithm (**left**) and the classic counterpart (**right**). The graphs were made using 2000 normal data images (decays that happened just as the particle entered the detector) and 2000 anomalous data images (decays that happened after the particle had traveled inside the detector).

For a better benchmark, Figure 9 shows the ROC curve and AUC comparisons between the classic autoencoder and the quantum model. The quantum algorithm did not reach the same level of performance as the classical equivalent. This was due to the reduced expressive power of the implemented quantum circuit, imposed by the constraints on the number of usable qubits and gates for the simulation. In any case, the quantum algorithm is already very close to the classical one and is expected to significantly improve with respect to the quantum hardware implementation, with the availability of the new generation of low-noise quantum circuits which has been announced to be available soon (see for example the IBM road map to quantum advantage: https://www.ibm.com/quantum/roadmap (accessed on 10 February 2023)).

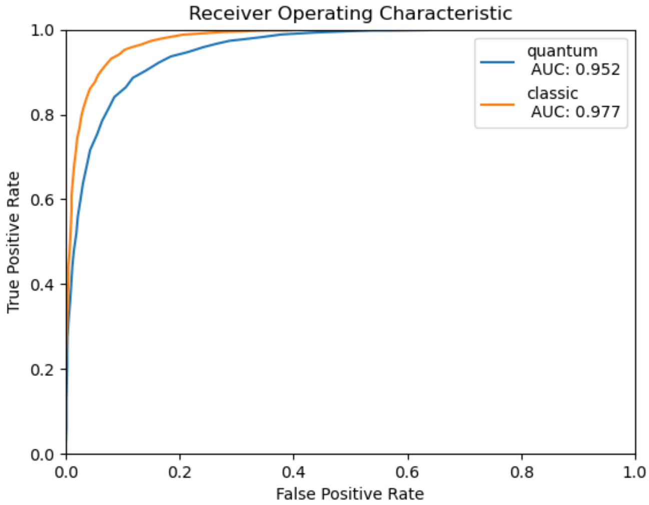


Figure 9. ROC curve and AUC for quantum anomaly detection algorithm (blue) and the classic counterpart (orange).

4. Test on Quantum Hardware

The execution of quantum circuits on NISQ devices, as stated in Section 3, is difficult even on state-of-the-art quantum devices. The main problems come from the high error rate of quantum gates, especially C-NOT gates that are fundamental to generate entanglement between qubits. Another important limitation comes from the connectivity in the architecture of quantum computers. In fact, it is not possible to apply C-NOT gates between all possible pairs of qubits. If an interaction between unconnected qubits is required, it is necessary to use SWAP gates to invert the quantum states of two qubits. As each SWAP gate is composed of three C-NOT gates [33], the use of noisy gates is further increased.

State-of-the-art quantum computers offer connectivity only between neighboring qubits, arranged in linear or circular structures. The architecture of the quantum computer used in this work (IBM_hanoi) is reported in Figure 10.

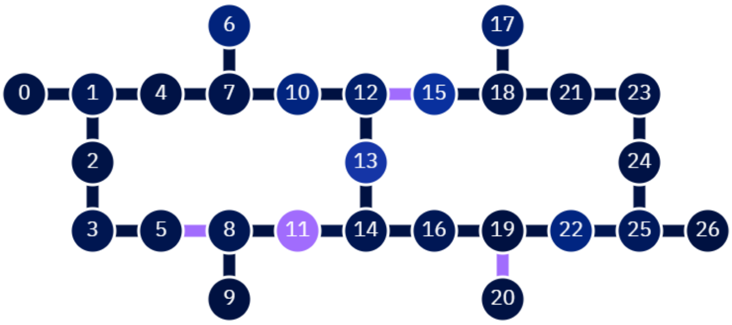


Figure 10. Architecture of the IBM_hanoi quantum computer. The C-NOT connectivity is reported. The colors of single qubits and their connections represent, respectively, the single qubit readout assignment error and C-NOT error probabilities. Darker colors represent a lower error probability, in a range between 5.9×10^{-3} and 9.8×10^{-2} for the readout error and 3.3×10^{-3} and 1 for C-NOT gates. Data from calibration on 19 October 2022.

With these limitations is impossible to achieve any significant performance on the IBM quantum hardware for the anomaly detection circuit we have developed. In order to make the algorithm work, some changes and a careful adaptation had to be implemented to reduce the complexity and size of the PQC. Given the consequent reduction in the expressive power of the model, we decided to focus only on the simplest use-case of the handwritten digits (see Section 3.1) for the quantum hardware test.

4.1. Adaptation to Quantum Hardware

We describe here the changes to the quantum circuit in order to solve the two problems discussed in the previous section, the amplitude encoding and the C-NOT connectivity.

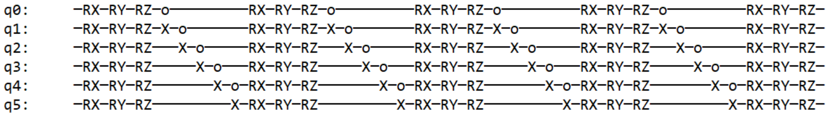
For the connectivity problem, it is important to notice that the PQC proposed in Section 2.2 requires only neighboring qubits interaction if the qubits are arranged in a circular topology. However, in our quantum hardware (Figure 10), it was not possible to find six qubits arranged in a ring. By removing the last C-NOT gate on each entangling layer, only neighboring qubits interactions were required for qubits arranged on a line. This allowed us to remove all SWAP gates from the circuit, thus reducing significantly the total number of C-NOT gates.

Moreover, we decided to use only four layers for the encoder circuit. With these changes, the expressive power of the PQC was reduced but still sufficient to detect anomalies. On noisy simulations, the performance of this PQC outperformed the one with six layers employed in Section 3.1.

Amplitude encoding is a state preparation procedure that allows one to transform the standard initial state (all qubits in the ground state) into a state that encodes the input data for the quantum algorithm. This procedure is necessary only to analyze classical data with quantum hardware. Implementing amplitude encoding on quantum hardware requires a number of C-NOT gates that grows exponentially in the number of qubits (Section 3). For a circuit of six qubits, more than one hundred C-NOT gates are required; this adds too much noise to the final result. To overcome the problem, we developed another PQC designed to provide a good approximation of the exact amplitude encoding while using a reduced number of gates. This circuit was trained to transform the initial ground state into a state that approximated the amplitude encoding. The parameters of the circuit were chosen by minimizing the mean squared error between the output state of the circuit and the target state (e.g., the correct amplitude encoding). For this procedure one circuit had to be trained for each normal or anomalous data element that we wanted to encode. The PQC ansatz was composed of four layers with the C-NOT gate disposition described before for the encoder (Figure 11). Moreover, a final layer composed only of rotation gates was added to increase expressive power and improve performance.

Each training was carried out for 15 epochs, with each epoch composed of 100 training steps, using the Adam optimizer and a learning rate of 0.01. In order to avoid barren plateaus, we started each training using the parameters of the previous PQC. Looking at the final loss values, we found out that the encoding of anomalous data (ones) was easier than the encoding of normal data (zeros).

In order to avoid a possible bias introduced by a bad amplitude encoding, we selected only the data encoded with a loss smaller than 0.1. We trained an approximated amplitude encoding with these characteristics for about 200 normal and 200 anomalous data.



```
q0:     ─RX─RY─RZ─o─────────RX─RY─RZ─o─────────RX─RY─RZ─o─────────RX─RY─RZ─o─────────RX─RY─RZ─
q1:     ─RX─RY─RZ─X─o───────RX─RY─RZ─X─o───────RX─RY─RZ─X─o───────RX─RY─RZ─X─o───────RX─RY─RZ─
q2:     ─RX─RY─RZ───X─o─────RX─RY─RZ───X─o─────RX─RY─RZ───X─o─────RX─RY─RZ───X─o─────RX─RY─RZ─
q3:     ─RX─RY─RZ─────X─o───RX─RY─RZ─────X─o───RX─RY─RZ─────X─o───RX─RY─RZ─────X─o───RX─RY─RZ─
q4:     ─RX─RY─RZ───────X─o─RX─RY─RZ───────X─o─RX─RY─RZ───────X─o─RX─RY─RZ───────X─o─RX─RY─RZ─
q5:     ─RX─RY─RZ─────────X─RX─RY─RZ─────────X─RX─RY─RZ─────────X─RX─RY─RZ─────────X─RX─RY─RZ─
```

Figure 11. Parametrized quantum circuit used for approximated amplitude encoding. The circuit was composed of four layers made of rotation gates and five C-NOT gates plus a final layer composed only of rotation gates.

4.2. Results

The final test on quantum hardware was carried out using a circuit composed of two parts, the approximated amplitude encoding circuit and the encoder (Section 4.1). The parameters of the encoder were trained with simulations on classic hardware in the same way as described in Section 3.1.

The parameters of the approximated amplitude encoding circuit were trained on each dataset as described in the previous section. In total, 200 samples of normal data and

200 samples of anomalous data were tested on the quantum hardware. Each circuit was executed with 2048 shots.

Figure 12 reports a comparison of the measurement counts of the circuit (only the first three qubits were measured) for a normal sample. On the left plot, the simulated circuit without noise is shown, and on the center plot, the simulated circuit with noise. For the noise model, we used the calibration values of IBM_hanoi taken on 19 October 2022 (the calibration can be accessed through IBM's qiskit library: https://qiskit.org (accessed on 10 February 2023)).

The plot on the right reports the counts for the circuit executed on the quantum hardware. As explained in Section 3.1, the loss function, which the circuit had to minimize, measured the sum of the probabilities of the three compressed qubits to be in the ground state. Thus, if the algorithm was working correctly, we expected the measured qubits to be mainly in the $|1\rangle$ state. The counts distribution was peaked, as expected, on the $|111\rangle$ state in all cases. The real quantum circuit, however, clearly showed a higher level of noise shown by higher counts for states different from $|111\rangle$. Moreover, the noise reported by IBM's simulation was lower than the one on real quantum hardware. This was probably due to the difficulty in reproducing a realistic noise model. In a real quantum circuit, there were many sources of noise, besides the readout measurement error and the gate error, so it was complicated to keep track of them all. Moreover, the noise parameters, obtained from calibration, changed over time. This made them no longer reliable if the circuit was not executed right after the calibration, as in this case.

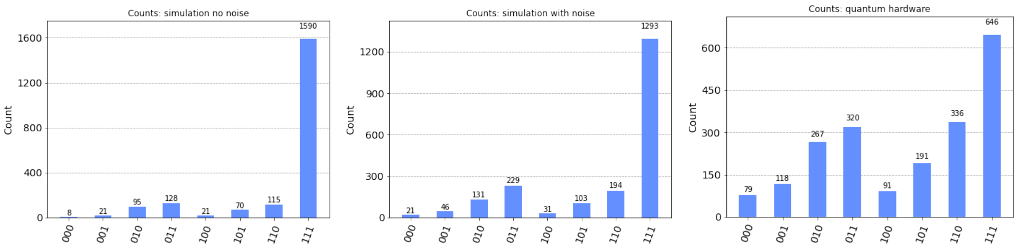


Figure 12. Counts distribution of 2048 shots for a simulated circuit with no noise (**left**) a simulated circuit with realistic noise (**center**) and a noisy quantum circuit (**right**).

The noise made the loss function distributions for normal and anomalous data almost indistinguishable. However, by observing the counts, we noticed that it was possible to improve the performance of the anomaly detection by using as loss function of one minus the probability of the $|111\rangle$ state. This loss function was different from the one previously employed. In fact, it can be easily observed that $1 - p(|111\rangle) \neq p(|0\rangle_1) + p(|0\rangle_2) + p(|0\rangle_3))$. This new loss function gave better results in the presence of noise because the state $|111\rangle$ was the one with the highest output probability. The output state probabilities were computed as the number of counts, for that specific state, divided by the total number of counts. Thus, the probabilities of states with higher counts were less affected by the noise, compared to the probabilities of states with a lower number of counts.

The distributions for normal and anomalous data for this loss function are reported in Figure 13 for simulated circuits with no noise (top left), simulated circuit with noise (top right) and real quantum circuit (bottom). For a better comparison, Figure 14 reports the ROC curves and the AUC for the three cases. It is possible to observe a significant separation between normal and anomalous data in the presence of noise, although with a clear degradation in the case of the execution on real quantum hardware (hardware vs. simulation AUC: 0.896 vs. 0.983).

It is interesting to notice that in the case of simulation, the ROC curves for the simulated circuits with or without noise almost overlapped even though the loss values for these two cases had different distributions. A low level of noise just shifted the losses to higher values but did not reduce the discriminative performance of the algorithm.

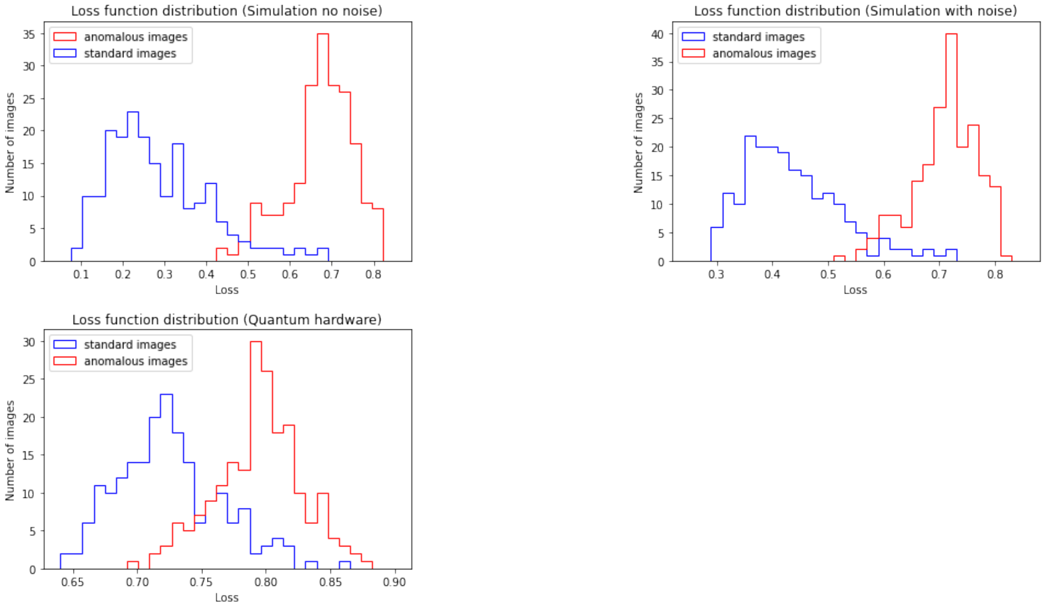


Figure 13. Quantum autoencoder loss function values distribution: simulated circuits with no noise (**top left**), simulated circuits with noise (**top right**) and a noisy quantum circuits (**bottom**). The graph was made using 200 normal data and 200 anomalous data, with 2048 shots each circuit. The loss function was one minus the probability of the $|111\rangle$ state.

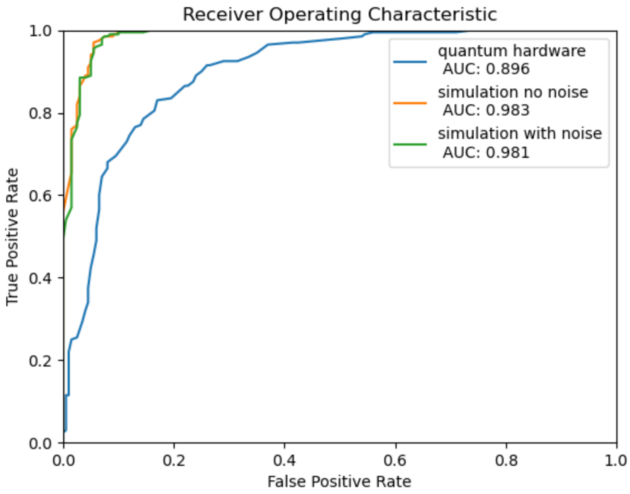


Figure 14. ROC curves and AUC for anomaly detection: simulated circuits with no noise (orange), simulated circuits with noise (green) and noisy quantum circuits (blue).

5. Conclusions

Quantum machine learning is a newborn topic with many possible algorithms still to be explored. In this work, we proved that it was possible to do anomaly detection for long-lived particles searches in a high-energy physics experiment using parametrized quantum circuits. Theoretically, without considering noise limitations, PQCs are powerful enough to distinguish anomalies in complex patterns such as the ones obtained in a muon spectrometer of a hadron collider experiment. With the currently available NISQ devices, it is only possible to execute very simple circuits on quantum hardware. These circuits must be adapted to the hardware limitations and can be used only for simpler tasks such as anomaly detection of handwritten digits. However, as quantum computers are improving rapidly [48,49], we expect, in the near future, to use PQCs to also solve complex tasks in particle physics. At that time, it will be possible to verify if, with fault-tolerant quantum computation, quantum machine learning outperforms classical machine learning algorithms on the analysis of classical data. Another possible direction to explore is the possibility to feed the quantum algorithms directly with quantum input data, thus avoiding the quantum data encoding step. Since the research and development of quantum sensing in particle detectors is a rapidly developing sector [50–52], this could lead to identify a possible advantage from the use of quantum machine learning algorithms of the type proposed in this work, in a not too distant future.

Author Contributions: Original idea and supervision, S.G.; conceptualization, S.B., D.S. and S.G.; methodology, S.B. and D.S.; software development, S.B., D.S. and T.S.; validation and data analysis, S.B., D.S. and T.S.; data curation, S.G.; writing, S.B., D.S., T.S., S.G. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Both the datasets and the code used for this study are available on request by contacting the authors.

Acknowledgments: Access to the IBM Quantum Services was obtained through the IBM Quantum Hub at CERN under sublicense agreement KR5386/IT. The views expressed are those of the authors and do not reflect the official policy or position of IBM. D.S. acknowledges Thales Alenia Space Italia for supporting the Ph.D. fellowship.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

QML	Quantum machine learning
NISQ	Noisy intermediate scale quantum
ANN	Artificial neural network
PQC	Parametrized quantum circuit
ROC	Receiver operating characteristic
CNN	Convolutional neural network
AUC	Area under the ROC curve
LHC	Large Hadron Collider
MDT	Muon drift chamber
RMS	Root mean square
ML	Machine learning

References

1. Jacob, B.; Peter, W.; Nicola, P.; Patrick, R.; Nathan, W.; Seth, L. Quantum machine learning. *Nature* **2017**, *549*, 195–202. [CrossRef]
2. García, D.P.; Cruz-Benito, J.; García-Peñalvo, F.J. Systematic Literature Review: Quantum Machine Learning and its applications. *arXiv* **2022**, arXiv:2201.04093.
3. Gregory, R.S.; Jonathan, P.O.; Dirk, E.; Jacques, C. Quantum optical neural networks. *Npj Quantum Inf.* **2019**, *5*, 60. [CrossRef]
4. Schuld, M.; Sinayskiy, I.; Petruccione, F. An introduction to quantum machine learning. *Contemp. Phys.* **2015**, *56*, 172–185.
5. Bank, D.; Koenigstein, N.; Giryes, R. Autoencoders. *arXiv* **2020**, arXiv:2003.05991.
6. Dong, G.; Liao, G.; Liu, H.; Kuang, G. A Review of the Autoencoder and Its Variants: A Comparative Perspective from Target Recognition in Synthetic-Aperture Radar Images. *IEEE Geosci. Remote Sens. Mag.* **2018**, *6*, 44–68. [CrossRef]
7. Hinton, G.E.; Salakhutdinov, R.R. Reducing the Dimensionality of Data with Neural Networks. *Science* **2006**, *313*, 504–507.
8. Cerezo, M.; Arrasmith, A.; Babbush, R.; Benjamin, S.C.; Endo, S.; Fujii, K.; McClean, J.R.; Mitarai, K.; Yuan, X.; Cincio, L.; et al. Variational quantum algorithms. *Nat. Rev. Phys.* **2021**, *3*, 625–644. [CrossRef]
9. Du, Y.; Hsieh, M.H.; Liu, T.; Tao, D. Expressive power of parametrized quantum circuits. *Phys. Rev. Res.* **2020**, *2*, 033125. [CrossRef]
10. Haug, T.; Bharti, K.; Kim, M. Capacity and Quantum Geometry of Parametrized Quantum Circuits. *PRX Quantum* **2021**, *2*, 040309. [CrossRef]
11. Gianelle, A.; Koppenburg, P.; Lucchesi, D.; Nicotra, D.; Rodrigues, E.; Sestini, L.; de Vries, J.; Zuliani, D. Quantum Machine Learning for b-jet charge identification. *J. High Energy Phys.* **2022**, *2022*, 14. [CrossRef]
12. Alvi, S.; Bauer, C.; Nachman, B. Quantum Anomaly Detection for Collider Physics. *arXiv* **2022**, arXiv:2206.08391.
13. Bauer, C.W.; Davoudi, Z.; Balantekin, A.B.; Bhattacharya, T.; Carena, M.; de Jong, W.A.; Draper, P.; El-Khadra, A.; Gemelke, N.; Hanada, M.; et al. Quantum Simulation for High Energy Physics. *arXiv* **2022**, arXiv:2204.03381.

14. Mott, A.; Job, J.; Vlimant, J.; Lidar, D.; Spiropulu, M. Solving a Higgs optimization problem with quantum annealing for machine learning. *Nature* **2017**, *550*, 375–379. [CrossRef] [PubMed]
15. Blance, A.; Spannowsky, M. Quantum machine learning for particle physics using a variational quantum classifier. *J. High Energy Phys.* **2021**, *2021*, 212. [CrossRef]
16. Terashi, K.; Kaneda, M.; Kishimoto, T.; Saito, M.; Sawada, R.; Tanaka, J. Event Classification with Quantum Machine Learning in High-Energy Physics. *Comput. Softw. Big Sci.* **2021**, *5*, 2. [CrossRef]
17. Chen, S.Y.C.; Wei, T.C.; Zhang, C.; Yu, H.; Yoo, S. Quantum Convolutional Neural Networks for High Energy Physics Data Analysis. Physical Review Research **2020**, *4*, 013231. [CrossRef]
18. Wu, S.L.; Chan, J.; Guan, W.; Sun, S.; Wang, A.; Zhou, C.; Livny, M.; Carminati, F.; Di Meglio, A.; Li, A.C.Y.; et al. Application of quantum machine learning using the quantum variational classifier method to high energy physics analysis at the LHC on IBM quantum computer simulator and hardware with 10 qubits. *J. Phys. G: Nucl. Part. Phys.* **2021**, *48*, 125003. [CrossRef]
19. Wu, S.L.; Sun, S.; Guan, W.; Zhou, C.; Chan, J.; Cheng, C.L.; Pham, T.; Qian, Y.; Wang, A.Z.; Zhang, R.; et al. Application of quantum machine learning using the quantum kernel algorithm on high energy physics analysis at the LHC. *Phys. Rev. Res.* **2021**, *3*, 033221. [CrossRef]
20. Bravo-Prieto, C.; Baglio, J.; Cè , M.; Francis, A.; Grabowska, D.M.; Carrazza, S. Style-based quantum generative adversarial networks for Monte Carlo events. *Quantum* **2022**, *6*, 777. [CrossRef]
21. Ngairangbam, V.S.; Spannowsky, M.; Takeuchi, M. Anomaly detection in high-energy physics using a quantum autoencoder. *Phys. Rev. D* **2022**, *105*, 095004. [CrossRef]
22. Bharti, K.; Cervera-Lierta, A.; Kyaw, T.H.; Haug, T.; Alperin-Lea, S.; Anand, A.; Degroote, M.; Heimonen, H.; Kottmann, J.S.; Menke, T.; et al. Noisy intermediate-scale quantum algorithms. *Rev. Mod. Phys.* **2022**, *94*, 015004. [CrossRef]
23. Coyle, B. Machine learning applications for noisy intermediate-scale quantum computers. *arXiv* **2022**, arXiv:2205.09414.
24. Arute, F.; Arya, K.; Babbush, R.; Bacon, D.; Bardin, J.C.; Barends, R.; Biswas, R.; Boixo, S.; Brandao, F.G.S.L.; Buell, D.A.; et al. Quantum supremacy using a programmable superconducting processor. *Nature* **2019**, *574*, 505–510. [CrossRef]
25. De Luca, G. A Survey of NISQ Era Hybrid Quantum-Classical Machine Learning Research. *J. Artif. Intell. Technol.* **2021**, *2*, 9–15. [CrossRef]
26. IBM Quantum. 2023. Available online: https://quantum-computing.ibm.com/ (accessed on 10 Feburay 2023) .
27. Huang, H.Y.; Broughton, M.; Mohseni, M.; Babbush, R.; Boixo, S.; Neven, H.; McClean, J.R. Power of data in quantum machine learning. *Nat. Commun.* **2021**, *12*, 2631. [CrossRef] [PubMed]
28. Evans, L.; Bryant, P. LHC Machine. *JINST* **2008**, *3*, S08001. [CrossRef]
29. Gneiting, T.; Vogel, P. Receiver Operating Characteristic (ROC) Curves. *arXiv* **2018**, arXiv:1809.04808.
30. Abohashima, Z.; Elhosen, M.; Houssein, E.H.; Mohamed, W.M. Classification with Quantum Machine Learning: A Survey. *arXiv* **2020**, arXiv:2006.12270.
31. Weigold, M.; Barzen, J.; Leymann, F.; Salm, M. Data Encoding Patterns for Quantum Computing. In Proceedings of the 27th Conference on Pattern Languages of Programs (PLoP '20), Virtual, 12–16 October 2020; The Hillside Group: Corryton, TN, USA, 2022.
32. Schuld, M.; Sweke, R.; Meyer, J.J. Effect of data encoding on the expressive power of variational quantum-machine-learning models. *Phys. Rev. A* **2021**, *103*, 032430. [CrossRef]
33. Benenti, G.; Casati, G.; Strini, G. *Principles of Quantum Computation and Information-Volume I: Basic Concepts*; World Scientific: Singapore , 2004.
34. Rojas, R. The Backpropagation Algorithm. In *Neural Networks: A Systematic Introduction*; Springer: Berlin/Heidelberg, Germany, 1996; pp. 149–182. [CrossRef]
35. Bravo-Prieto, C. Quantum autoencoders with enhanced data encoding. *Mach. Learn. Sci. Technol.* **2021**, *2*, 035028. [CrossRef]
36. Efthymiou, S.; Ramos-Calderer, S.; Bravo-Prieto, C.; Pérez-Salinas, A.; García-Martín, D.; Garcia-Saez, A.; Latorre, J.I.; Carrazza, S. Qibo: A framework for quantum simulation with hardware acceleration. *Quantum Sci. Technol.* **2021**, *7*, 015018. [CrossRef]
37. Abadi, M.; Agarwal, A.; Barham, P.; Brevdo, E.; Chen, Z.; Citro, C.; Corrado, G.S.; Davis, A.; Dean, J.; Devin, M.; et al. TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems. *arXiv* **2015**, arXiv:1603.04467.
38. McClean, J.R.; Boixo, S.; Smelyanskiy, V.N.; Babbush, R.; Neven, H. Barren plateaus in quantum neural network training landscapes. *Nat. Commun.* **2018**, *9*, 4812. [CrossRef] [PubMed]
39. Cerezo, M.; Sone, A.; Volkoff, T.; Cincio, L.; Coles, P.J. Cost function dependent barren plateaus in shallow parametrized quantum circuits. *Nat. Commun.* **2021**, *12*, 1791. [CrossRef]
40. Shende, V.V.; Bullock, S.S.; Markov, I.L. Synthesis of quantum-logic circuits. *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.* **2006**, *25*, 1000–1010. [CrossRef]
41. Kingma, D.P.; Ba, J. Adam: A Method for Stochastic Optimization. *arXiv* **2014**, arXiv:1412.6980.
42. Falkowski, A.; Ruderman, J.A.T.; Volansky, V.T.; Zupan, J. Hidden Higgs decaying to lepton jets. *J. High Energy Phys.* **2010**, *2010*, 77. [CrossRef]
43. Strassler, M.J.; Zurek, K.M. Echoes of a hidden valley at hadron colliders. *Phys. Lett. B* **2007**, *651*, 374–379. [CrossRef]
44. ATLAS Collaboration. The ATLAS Experiment at the CERN Large Hadron Collider. *JINST* **2008**, *3*, S08003. [CrossRef]
45. ATLAS Collaboration. Standalone vertex finding in the ATLAS muon spectrometer. *JINST* **2014**, *9*, P02001. [CrossRef]

46. ATLAS Collaboration. Search for long-lived neutral particles produced in pp collisions at $\sqrt{s} = 13$ TeV decaying into displaced hadronic jets in the ATLAS inner detector and muon spectrometer. *Phys. Rev. D* **2020**, *101*, 052013. [CrossRef]
47. Skolik, A.; McClean, J.R.; Mohseni, M.; van der Smagt, P.; Leib, M. Layerwise learning for quantum neural networks. *Quantum Mach. Intell.* **2021**, *3*, 5. [CrossRef]
48. Ací n, A.; Bloch, I.; Buhrman, H.; Calarco, T.; Eichler, C.; Eisert, J.; Esteve, D.; Gisin, N.; Glaser, S.J.; Jelezko, F.; et al. The quantum technologies roadmap: A European community view. *New J. Phys.* **2018**, *20*, 080201. [CrossRef]
49. Gill, S.S.; Kumar, A.; Singh, H.; Singh, M.; Kaur, K.; Usman, M.; Buyya, R. Quantum Computing: A Taxonomy, Systematic Review and Future Directions. *arXiv* **2020**, arXiv: 2010.15559.
50. Kaltenbaek, R.; Acin, A.; Bacsardi, L.; Bianco, P.; Bouyer, P.; Diamanti, E.; Marquardt, C.; Omar, Y.; Pruneri, V.; Rasel, E.; et al. Quantum technologies in space. *Exp. Astron.* **2021**, *51*, 1677–1694. [CrossRef]
51. Bass, S.D.; Erez, Z. Quantum technologies in particle physics. *Phil. Trans. R. Soc.* **2021**, *380*, 20210072. [CrossRef]
52. Doser, M.; Auffray, E.; Brunbauer, F.; Frank, I.; Hillemanns, H.; Orlandini, G.; Kornakov, G. Quantum Systems for Enhanced High Energy Particle Physics Detectors. *Front. Phys.* **2022**, *10*, 483. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.