LIP-STUDENTS-23-16

Quantum Machine Learning applied to HEP: a Pragmatic Approach

Bruna Salgado[1,a] and Catarina Felgueiras[1,b]

[1] *LIP and Universidade do Minho, Braga, Portugal*

Project supervisors: Nuno Filipe Castro, Maria Gabriela Oliveira, Miguel Peixoto

October 17, 2023

Abstract. This study aims to provide insights into the potential benefits and challenges of quantum computing for classification tasks in High Energy Physics, by evaluating Quantum Machine Learning and Classical Machine Learning methods on a comparable way. To attain this goal, a grid search was conducted to identify the optimal parameters for both the Variational Quantum Classifier and a classical shallow machine learning method, to ensure a fair and unbiased comparison between the two models. Finally, a Variational Quantum Classifier model was executed on a real quantum computer to evaluate its performance under practical quantum computing conditions.

Keywords: High Energy Physics, Variational Quantum Classifier, Support Vector Machine

1 Introduction

The Standard Model of Particle Physics (SM) has achieved considerable success in explaining the fundamental building blocks of matter and their interactions. Nonetheless, significant questions remain, including gravity, dark matter, dark energy, and the imbalance between matter and antimatter in the universe. This drives the the ongoing quest for new physics beyond the SM (BSM) at CERN's Large Hadron Collider (LHC) [1].

The exploration of BSM phenomena at colliders presents challenges due to vast datasets and low signal-to-background ratios. To tackle this, machine learning (ML) techniques, particularly for classification tasks, have been employed, revealing their remarkable ability to identify correlations in high-dimensional parameter spaces [1].

In this study, a systematic comparison is made between the performance of Quantum Machine Learning (QML) and shallow Classical Machine Learning (CML) algorithms in the context of High Energy Physics (HEP). The primary focus is on binary classification tasks, specifically distinguishing between BSM signals and SM background. The investigation involves the utilization of Variational Quantum Classifiers (VQC) optimized through a grid search, while also exploring the potential of reduced data through feature reduction techniques.

Brief Overview

In this article, the VQC and its implementation are discussed in detail in section 2. Subsequently, Support Vector Machines (SVMs) are explained in section 3. Following this, the dataset utilized is briefly outlined in section 4. We then proceed to explore Sequential Feature Selection methods, including SBS and PCA, as detailed in section 5. The optimal SVM and VQC models are identified through a grid search in section 6. Finally, the VQC model is executed on a real quantum computer and compared with the optimal SVM model in section 7.

[a] e-mail: pg50266@uminho.pt

[b] e-mail: a100506@uminho.pt

2 Variational Quantum Classifier

Within the scope of this research, the VQC is employed as a binary classifier. The dataset consists of events, each associated with a label ('signal' or 'background'), representing the two classes. The main objective of the VQC is to learn a mapping between the input data and their corresponding labels, allowing it to accurately predict the class of new, unseen events.

The VQC employs a series of quantum gates to form a parameterized quantum circuit. These gates are characterized by adjustable parameters (w) that can be tuned during the training process.

The binary classification process using the VQC involves four key stages: data embedding, ansatz, final prediction, and classical optimization.

2.1 Data embedding

The process of data embedding entails converting classical data into a quantum state, denoted as $|\psi_X\rangle$. Several methods have been advanced for this purpose [2], and following empirical assessment, angle embedding has exhibited superior performance when compared to amplitude embedding [3]. Consequently, angle embedding has been selected as the preferred approach.

For a classical information vector of dimension N, denoted as $X = (x_1, x_2, ..., x_N)$, the encoded state is determined through the application of a state preparation circuit to the initial ground state $|0\rangle^{\otimes N}$. Each element of the vector X is encoded as an angle, requiring the utilization of N qubits to accommodate the N features from the original dataset. This is accomplished by applying a rotation gate to each individual qubit. In the context of this study, data embedding is performed through rotation around the x-axis on the Bloch sphere.

It is important to denote that angle embedding requires each feature to be normalized within the range of $[-\pi, \pi]$.

2.2 Ansatz

The ansatz corresponds to a parameterized quantum circuit. The VQC utilizes a ansatz denoted as $U(w)$, where

the learning parameters (w) play a crucial role in evolving a quantum state embedded with classical information (X) into the final state ($|\psi'_X\rangle$). The model circuit is composed of layers comprising rotation gate (R) and CNOT gates to entangle the qubits. The rotation gate, represented as $R(\phi, \theta, \omega)$, possesses three adjustable parameters allowing it to rotate any arbitrary state to any location on the Bloch sphere.

The weight vector (w) encompasses all learning parameters, structured with dimensions ($n \times l$ x 3), where n denotes the number of qubits, and l represents the number of layers. Subsequent to the application of rotation gates, CNOT gates are employed to to entangle the qubits based on control and target qubit values, following Algorithm 1.

Algorithm 1

Requires: $n \geq 2$, where n is the number of qubits
  if $n == 2$ then
    $CNOT(1, 0)$
  else
    for qubit $\leftarrow$ 0 to $n - 1$ do
      if $qubit == n - 1$ then
        $CNOT(qubit, 0)$
      else
        $CNOT(qubit, qubit + 1)$
      end if
    end for
  end if

Algorithm 1. CNOT Arrangement.

2.3 Final Prediction

During the final prediction stage, an observable measurement is performed on one (or more) of the qubits in the state $|\psi'_X\rangle$. This measurement provides the model's prediction for the binary classification task.

The result/prediction is achieved by evaluating the expected value of the Pauli operator $\hat{\sigma}_z$ on one of the qubits in the final state $|\psi'_X\rangle$.

The choice to measure only one of the qubits contemplates the problem known as "barren plateaus" or "vanishing gradients," where, under certain conditions, the gradient of the cost function exponentially diminishes as the system size increases. Employing local observables leads to, at worst, a polynomially vanishing gradient. [4, 5].

2.4 Classical Optimization

It is important to emphasize that VQC training is directed towards identifying the optimal optimal set of learning parameters w. During the training process, a classical optimizer is employed to identify the optimal learning parameters that minimize a specific cost function, in this case, the square loss. This cost function serves as a quantitative measure of the discrepancy between the output generated by the VQC and the desired target.

The optimization procedure follows an iterative feedback loop, where the cost function is evaluated at each iteration, and the resulting updated parameters are fed back into the circuit for subsequent refinement. This iterative process continues until a satisfactory convergence criterion is met, leading to the determination of the most suitable circuit parameters that best approximate the target quantum state.

The following diagram illustrates the VQC described.

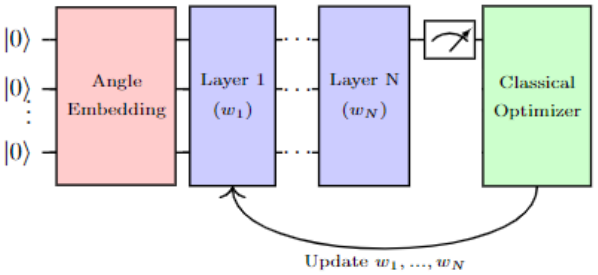

Figure 1. Variational Quantum Classifier (VQC) structure.

Figure 2 corresponds to the quantum circuit of the VQC obtained in Pennylane for 3 features and 2 layers.

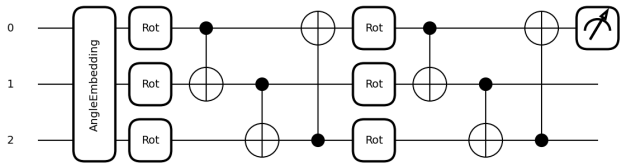

Figure 2. An example circuit for the VQC architecture used. It is comprised of 2 layers and 3 features.

2.5 Implementation details

The training commences by initializing the weight vector randomly, typically within an order of magnitude of 10^{-2}. Subsequently, it undergoes multiple training iterations until reaching the maximum specified epochs or until the validation AUC score stabilizes - early stopping. The condition for early stopping was defined as not achieving a superior AUC score to the previous best AUC score in 20 epochs. In each iteration, the model is applied to the training dataset, computing the cost function, and then updating the model parameters using the Adam optimizer [6].

The implemented algorithm is summarized bellow.

Algorithm 2

```
params ← params_initialization()
for epoch ← 1 to max_epochs do
    loss ← cost(params)
    params ← optimizer.step()
    if epoch_number%5 == 0 or epoch_number
    == max_epochs then
        validation_step()
    end if
    if early_stopping_cond and epoch_number >
    min_epochs then
        break
    end if
end for
```

Algorithm 2. VQC implementation.

2.6 Characteristics of the simulation

The quantum machine learning experiments were simulated in Pennylane's default.qubit quantum simulator [7] to conduct a grid search for discovering the optimal model. Subsequently, the top-performing model's performance was evaluated on a real quantum computer using PennyLane's integration with IBM's quantum computing framework, Qiskit [8].

3 The Classical Machine Learning Aproach: SVM

The QML model is assessed against a shallow CML technique, specifically the Support Vector Machine (SVM), which is chosen as the baseline for comparison.

An SVM classifier is trained to delineate two classes of data in the feature hyperspace by identifying the optimal hyperplane that effectively separates them. This is achieved by utilizing support vectors, which are data points located closest to the hyperplane within the two classes.

The loss function of the SVM is designed to maximize the margin, representing the distance between the hyperplane and the nearest data point from either class. The primary objective is to identify the hyperplane with the greatest possible distance to the nearest data point from either class within the training dataset, ultimately enhancing the classification of new data points.

This method was implemented using scikit-learn [9].

4 Dataset

This study involves the analysis of a dataset that encompasses simulated events resulting from pp collisions at 13 TeV, as outlined in [10]. These events correspond to final states characterized by the presence of 2 leptons, at least 1 b-jet, at least 1 large-R jet, and a significant scalar sum of transverse momentum (pT) for all reconstructed particles within the event (HT > 500 GeV). These attributes represent a prevalent topology frequently employed in LHC investigations focused on Beyond the Standard Model (BSM) events. The primary objective is to differentiate the predominant Standard Model (SM) background, $Zb\bar{b}$, from a BSM signal linked to top-quark pair production with a flavor-changing neutral current decay (t $\rightarrow$ qZ, where q = c, u). The chosen BSM signal exhibits kinematic similarities with the background, rendering it a suitable benchmark for assessment.

Both classical and quantum ML algorithms are trained using a comprehensive set of features, including properties of leading jets, large-R jets, leptons, missing transverse energy (MET), and multiplicities of various particles. The training process and computation of metrics incorporates Monte Carlo weights based on theoretical predictions at a target luminosity of 150 fb^{-1}.

5 Feature Selection

The dataset encompasses a total of 47 features. Given the specific data embedding method employed, training a Variational Quantum Classifier (VQC) using all dataset features would require the utilization of 47 qubits. However, this exceeds the practical capabilities of currently available quantum computers, rendering it infeasible to train a VQC with the complete feature set. Consequently, only quantum computers with a maximum of 5 qubits were considered for analysis. In this study, two feature selection methods were explored: Sequential Feature Selection (SFS) and Principal Component Analysis (PCA).

It is noteworthy to mention that the metric used in this context is the AUC-ROC.

5.1 Sequential Feature Selection

SFS methods aim to identify the most relevant features for a given problem. Among these methods, the Sequential Backward Selection (SBS) algorithm stands as a variation of the SFS approach. SBS initiates with the complete feature set and iteratively removes one feature at a time based on classifier performance until the desired feature subset size, k, is reached.

This feature selection method is implemented to the dataset considered in [1] . The authors deemed it appropriate to focus on non-discrete variables, as they found these variables to significantly reduce the variability of the outcomes. The corresponding results are depicted in Table 1

Table 1. Features selected by the SBS Algorithm and their respective AUC Score on the training dataset.

Feature	AUC
$\not{E}_T$	0.817
large R-jet τ_1	0.576
large R-jet τ_3	0.316
Jet$_2$ p_T	0.313
Jet$_1$ p_T	0.292

5.2 Principal Component Analysis

In this study, PCA is employed to reduce the dimensionality of a highly correlated, high-dimensional dataset while ensuring the resulting features are uncorrelated. In the context of this study, the primary goal of PCA is to eliminate feature correlations while preserving the original data's dimensionality.

The PCA transformation is learned from the training dataset and consistently applied to all datasets. PCA components are ranked based on their AUC scores. The study selects the top 5 ranked PCA components, which are presented in Table 2. The implementation of PCA from scikit-learn is utilized to perform this analysis [9].

Table 2. Features selected by the PCA Algorithm and their respective AUC Score on the training dataset.

Component	AUC
Component 1	0.775146
Component 3	0.715941
Component 0	0.687727
Component 14	0.630145
Component 36	0.605685

6 Finding the Optimal Models

A comprehensive grid search was executed to ascertain the optimal SVM and VQC models. Due to temporal constraints, a preliminary grid search was conducted, focusing on model-specific hyperparameters, and limited to a 2-component/feature model along with a dataset containing 1000 data points.

6.1 SVM

Regarding the SVM gridsearch, the model-specific hyperparameter sets considered along with the corresponding results are presented in Table 3.

Table 3. List of scanned SVM hyperparameters for the preliminary grid search.

Variable HP	Possible Values	Optimal value: PCA	Optimal value: SBS
C	[0.0001, 0.001, 0.01, 0.1, 1, 10, 100, 1000]	1000	1000
gamma	[0.0001, 0.001, 0.01, 0.1, 1, 10, 'scale']	0.001	0.0001
kernel	['linear', 'rbf', 'poly']	'rbf'	'rbf'

Subsequently, for each hyperparameter set, five models were trained on five distinct subsets of the initial dataset, employing random sampling. The sets of hyperparameters under consideration in this second grid search, are summarized in Table 4.

Table 4. List of scanned SVM hyperparameters

Variable Hp	Possible Values
Feature Selection	[PCA, SBS]
Number of Data points	[250, 500, 1k, 2k, 4k]
Number of Features	[1, 2, 3, 4, 5]

6.1.1 Results

The results of the grid search corresponding to Table 4 are presented in Figure 3. Furthermore, an ensemble method was applied.

This method consists of training the multiple individual models to enhance overall predictive performance. By aggregating the outputs of these base models, an ensemble method aims to mitigate individual model weaknesses and produce more robust and accurate predictions. In this instance, 5 SVM models were trained using 5 distinct training datasets. Following this, each model was utilized to classify the test dataset. The AUC score was then calculated using the mean of the individual output scores produced by these 5 models.

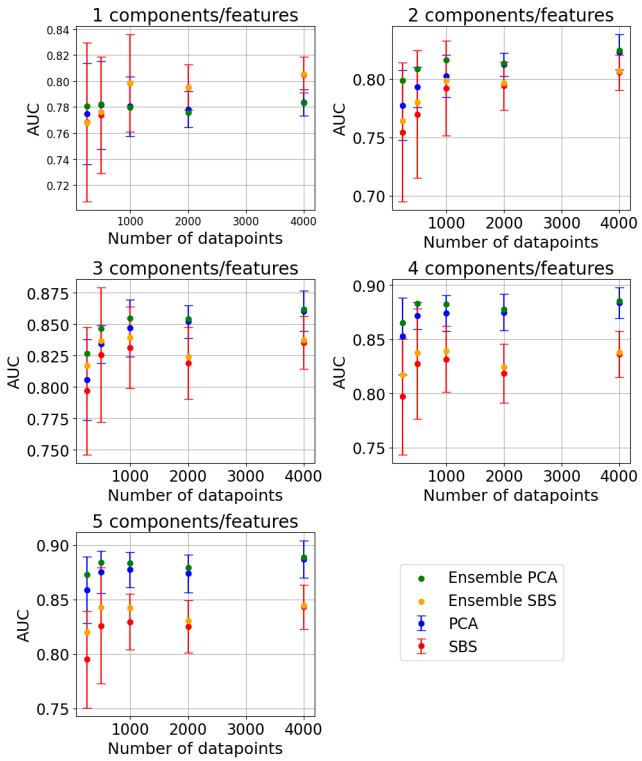

Figure 3. Plot grid representing the results for the SVM grid search. Each data point represents the AUC score on the test dataset of a different set of HP, as listed in Table 4. The error bar represents the standard deviation associated with each data point, where each point is the average of five different random samplings from the data.

Observing Figure 3, it becomes evident that the performance achieved with 3, 4, and 5 components/features is compatible. Notably, the highest AUC is attained with PCA for 5 components and 4000 datapoints, indicating this model as the most probable optimal choice.

Regarding the comparative performance between PCA and SBS, PCA generally outperforms SBS. However, for 1000, 2000, and 4000 datapoints with 1 component/feature, SBS yields the highest AUC. Moreover, within PCA, the ensemble method consistently produces superior results. Additionally, the analysis of the outcomes suggests that PCA generates more stable results, especially when dealing with fewer datapoints.

The analysis of the ensemble results reveals that, in general, the ensemble method consistently outperforms the non-ensemble approach in both PCA and SBS. Furthermore, this performance gap tends to be more pronounced in scenarios with fewer datapoints. In addition, the ensemble results are in accord with the error bars of the respective non-ensemble results.

6.2 Variational Quantum Classifier

Next, an analogous process was employed to determine the optimal VQC model. The preliminary grid search involved evaluating a set of learning rates, and the corresponding results are detailed in Table 5.

Table 5. List of considered VQC learning rates for the preliminary gridsearch

Variable HP	Possible Values	Optimal value: PCA	Optimal value: SBS
Learning Rate	[0.001, 0.005, 0.01, 0.03, 0.05, 0.1, 0.5, 1.0]	0.5	0.5

Afterward, a grid search was conducted, utilizing the optimal learning rate previously obtained for each feature selection technique considered. However, due to technical and time-related issues, it was not possible to conclude the grid search by the time this report had to be submitted.

7 Real Quantum Computers Results

It is now necessary to access the performance of the quantum algorithm in a real quantum computer. The quantum model adopted utilizes PCA as the feature selection technique with 5 components, 5 layers, and 4000 data points. The final expectation value for each event was determined by averaging over 20,000 shots conducted on the quantum computer. The *ibm_belem* was the quantum device chosen to perform the quantum experiments.

The comparison of the performance achieved with SVM, VQC in simulation, VQC in real device, and VQC in simulation with quantum device noise is illustrated in Figure 4. Additionally, Figure 5 presents the variability of the ROC curve for each model/simulation and the respective AUC scores are summarized in Table 6.

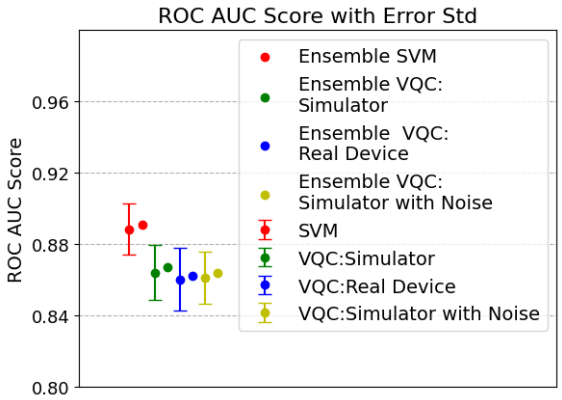

Figure 4. Comparison of the performance achieved with SVM, VQC in simulation, VQC in a real device (*ibm_belem*), and VQC in simulation with quantum noise from *ibm_belem*. The error bar represents the standard deviation associated with each data point, where each point is the average of five different random samplings from the original dataset.

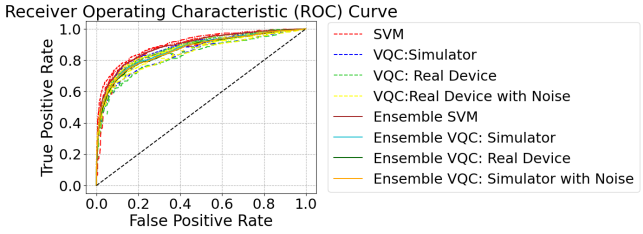

Figure 5. Variability of the ROC curve obtained for the SVM, VQC in simulation, VQC in a real device (*ibm_belem*), and VQC in simulation with quantum noise from *ibm_belem*.

Table 6. AUC scores of each sample tested.

Sample	SVM	VQC: Simulation	VQC: Real Device	VQC: Sim With Noise
Sample 1	0.9	0.88	0.88	0.87
Sample 2	0.9	0.87	0.87	0.86
Sample 3	0.89	0.88	0.88	0.88
Sample 4	0.86	0.84	0.83	0.84
Sample 5	0.89	0.86	0.85	0.86
Ensemble	0.89	0.87	0.86	0.86

Analysing Figure 4, Figure 5 and Table 6, it is noticeable that the SVM and VQC results are compatible. It is important to reiterate that these results do not stem from a fair comparison due to the suboptimal hyperparameters of the VQC model, as discussed earlier.

Observing Figure 4, it is concluded that the ensemble method results are in accordance with the variability and are, in general, slightly better than the mean of the results of the 5 samples.

Regarding the comparison between simulation and real-device implementation, it is concluded that the results are very compatible.

8 Conclusion

The primary purpose of this study is to explore the application of QML on HEP datasets.

Initially, a grid search was conducted to identify the optimal classical and quantum models, which encompassed the consideration of two feature selection techniques: SBS and PCA. The findings relative to the SVM grid search indicate that PCA, generally, yields better and more stable results.

Then, the quantum model was tested in a real quantum computer, and it was found that the simulated, real device and SVM results were compatible. Additionally, the results suggest that the ensemble method exhibits superior performance when compared with the mean of the AUC Score of the 5 samples.

In conclusion, we found no evidence of a quantum advantage when handling HEP datasets. However, it's important to note that the quantum model used may be suboptimal, so future work might still shed some light into this.

Future work would focus on exploring novel architectures for the VQC to enhance its overall performance. Furthermore, a comprehensive grid search encompassing SVM-specific parameters and learning rates for the VQC should be conducted, considering the entire set of datapoints and features previously considered.

9 Acknowledgements

We extend our heartfelt gratitude to our supervisors, Professor Nuno Castro, Gabriela Oliveira, and Miguel Peixoto, for their invaluable guidance and assistance throughout the development of this project. We are also thankful to the University of Minho and LIP for providing us with the opportunity and workspace to work on this project.

References

[1] M.C. Peixoto, N.F. Castro, M.C. Romão, M.G.J. Oliveira, I. Ochoa, arXiv preprint arXiv:2211.03233 (2022)

[2] R. LaRose, B. Coyle, Physical Review A **102**, 032420 (2020)

[3] A. Gianelle, P. Koppenburg, D. Lucchesi, D. Nicotra, E. Rodrigues, L. Sestini, J. de Vries, D. Zuliani, Journal of High Energy Physics **2022**, 1 (2022)

[4] Z. Holmes, K. Sharma, M. Cerezo, P.J. Coles, PRX Quantum **3**, 010313 (2022)

[5] M. Cerezo, A. Sone, T. Volkoff, L. Cincio, P.J. Coles, Nature communications **12**, 1791 (2021)

[6] D.P. Kingma, J. Ba, arXiv preprint arXiv:1412.6980 (2014)

[7] V. Bergholm, J. Izaac, M. Schuld, C. Gogolin, S. Ahmed, V. Ajith, M.S. Alam, G. Alonso-Linaje, B. AkashNarayanan, A. Asadi et al., arXiv preprint arXiv:1811.04968 (2018)

[8] M.S. Anis, H. Abraham, R.A. AduOffei, G. Agliardi, M. Aharoni, I.Y. Akhalwaya, G. Aleksandrowicz, T. Alexander, M. Amy, S. Anagolum et al., Qiskit/qiskit (2021)

[9] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg et al., the Journal of machine Learning research **12**, 2825 (2011)

[10] M. Crispim Romao, N.F. Castro, R. Pedro, *Simulated pp collisions at 13 TeV with 2 leptons + 1 b jet final state and selected benchmark Beyond the Standard Model signals* (2021), https://doi.org/10.5281/zenodo.5126747