Gamma-hadron Separation in Imaging Atmospheric Cherenkov Telescopes using Quantum Classifiers

Jashwanth S[2], Sudeep Ghosh[1], Kavitha Yogaraj[1], Neha Shah[2], and Ankhi Roy[3]

[1]IBM Quantum, Bengaluru, Karnataka,India.
[2]Department of Physics, Indian Institute of Technology Patna, Bihar, India.
[3]Department of Physics, Indian Institute of Technology Indore, Madhya Pradesh, India.

October 11, 2022

Abstract

In this paper we have introduced a novel method for gamma hadron separation in Imaging Atmospheric Cherenkov Telescopes(IACT) using Quantum Machine Learning. IACTs captures images of Extensive Air Showers (EAS) produced from very high energy gamma rays. We have used the QML Algorthims, Quantum Support Vector Classifier (QSVC) and Variational Quantum Classifier(VQC) for binary classification of the events into signals(Gamma) and background(hadron) using the image parameters. MAGIC Gamma Telescope dataset is used for this study which was generated from Monte Carlo Software Coriska. These quantum algorithms achieves performance comparable to standard multivariate classification techniques and can be used to solve variety of real world problems. The classification accuracy is improved by hyper parameter tuning. We propose a new architecture for using QSVC efficiently on large datasets and found that clustering enhance the overall performance.

1 Introduction

Imaging Atmospheric Cherenkov Technique (IACT) is a ground based gamma ray observation technique, used in γ ray astrophysics to detect very high energy γ ray photons[1]. The Earth's atmosphere serves as a giant local calorimeter for gamma ray detection. When a very high energy gamma ray enters the atmosphere, it interacts with atmospheric nuclei and an e^+e^-pair is produced. The produced e^+ and e^- from the pair will make collisions with atmospheric nuclei and are subjected to energy loss via multiple Coulomb scattering. In these collisions the charged particles will be accelerated and emit electromagnetic radiation (bremsstrahlung). This process continues till the threshold for the physical processes is involved is reached and an electromagnetic cascade is produced, also known as air shower[1]. The charged particles in this air shower are travelling faster than the phase velocity of light in the atmosphere and will emit Cherenkov radiations at Cherenkov angle, which is inversely proportional to the velocity of the particle and the refractive index[2]. A very narrow cone (1.0° at 8 km above sea level) of Cherenkov radiation generated by the ultrarelativistic charged particle in the cascade penetrates to the ground level[3]. The produced Cherenkov light is spread over a larger area of several hundred square meter around the shower axis and can be detected using ground based telescopes working on the principle of IACT. An arrangement of large focusing mirrors reflects the Cherenkov light onto an array of photo-multiplier tubes which captures high speed images of these very short Cherenkov radiation flashes (5-20 ns)[4]. The direction and energy of the primary gamma rays can be reconstructed from the shower image using different techniques.

An overwhelming background of hadronic cascade is initiated by the cosmic rays entering into Earth's atmosphere. To study high energy gamma rays of astrophysical origin, the intense isotropic background of hadronic cascade from cosmic rays must be rejected with high efficiency without losing much of the primary gamma ray signal. Traditionally, a direct selection method is used for the γ-hadron separation in Cherenkov telescope data analysis, in which direct cuts are made on the image parameters[5]. The γ-hadron separation is a multivariate binary classification of the events into two classes : signal (gamma) and background (hadrons) using different shape features of the shower image. A huge class imbalance is present in the Cherenkov telescope data due to the dominant hadronic background. The number of gamma ray signals are very small compared to hadronic events, not only that the exact signal to background ratio is also unknown and may vary for different sources[6]. A mix of Monte Carlo generated gamma ray events and real hadronic background can be used for training purpose in the classical machine learning algorithms[6]. Several classical machine learning algorithms, Random forest[7], and deep learning methods, Convolutional Neural Networks(CNNs)[1, 8], are found very effective in extracting the gamma rays signal from a dominant hadronic background. Due to the ever changing nature of the Earth's atmosphere, which acts as a calorimeter for incoming rays, the air showers produced by both primary gamma rays and cosmic rays are subjected to fluctuations[9]. Hadronic showers show greater shower-to-shower fluctuations in its longitudinal profile. In addition, light from local particles at the tail of shower will cause intensity fluctuations close to the shower axis. Therefore, there is finite chance of misclassification of events.

In 1982 Richard Feynman come up with the idea that nature is not classical and hence it is better to use quantum computers to make a simulation of nature[10]. The field of Quantum Computing is advancing at high pace and new computational techniques are being developed and tested to solve problems in various areas of physics. In this letter, we are exploring use of one such technique, Quantum Machine Learning (QML), to study the gamma-hadron separation in the IACT Telescopes. The result obtained are based on the Variational Quantum Classifier [11] and Quantum Support Vector Machine (QSVM)[12].

2 Gamma hadron separation

The gamma rays with energies ranging from $100GeV$ to $10TeV$ can be separated efficiently from a hugely dominant hadronic background using several image parameters based on the shape and orientation of the shower images. At very high energies the hadronic shower gets narrower and less different from gamma ray induced showers,except that an increased muon background will be produced. But these muons don't have much role in $\gamma-$hadron separation.

2.1 Lateral and longitudinal developments of air showers

Lateral distribution of light in gamma ray showers is be governed by the cherenkov angle and multiple Coulomb scattering.Half of the total cherenkov light of gamma cascade is emitted from the shower core which results in a peak in the lateral distribution which is absent in case of hadronic shower[13]. This difference in the lateral distribution of cherenkov light at the ground level is crucial in γ-hadrons separation. Moreover the hadronic shower images are much wider whereas the gamma events has a more regular shape. If we analyse the longitudinal and lateral development of showers, hadron showers having higher transverse momentum flow to the pions has greater transverse development throughout the shower compared to that of a gamma shower[14]. Energetic muons produced from the decay of charged pions will reach the ground over a wide region and produces local cherenkov peaks. The hadron showers with cherenkov producing components much closer to the ground, shows more fluctuations in the cherenkov emission and longitudinal development compared to gamma showers. The gamma ray showers being more of an electromagnetic cascade has greater longitudinal development[5].

2.2 Image parametrization

The hadronic shower images are observed to be more longer and broader compared to gamma ray shower images. Effective discrimination of primary gamma ray shower and hadronic background shower is possible on the basis of the width,length and orientation of these images. The showers with axis parallel to the optic axis and landing directly on the detector will produce circular images. But if it lands at some distance(impact parameter) away from the detector it will be a bivariate Gaussian distribution which is an elliptical cluster. For gamma ray showers the major axis is oriented towards the camera center whereas the hadronic showers have random axial orientation.

The pixel image of the shower after some pre processing and image cleaning is then converted to into few image parameters, defined by moment analysis on the pixel signal amplitudes. The moments are defined as:

$$
\begin{aligned}
\langle x\rangle &= \frac{\Sigma n_i x_i}{\Sigma n_i} \quad \langle y\rangle = \frac{\Sigma n_i y_i}{\Sigma n_i} \quad \langle xy\rangle = \frac{\Sigma n_i x_i y_i}{\Sigma n_i} \\
\langle x^2\rangle &= \frac{\Sigma n_i x_i^2}{\Sigma n_i} \quad \langle y^2\rangle = \frac{\Sigma n_i y_i^2}{\Sigma n_i}
\end{aligned}
\tag{1}
$$

where, x and y are the coordinates of pixels and n is the number of digital counts in a pixel. The summation runs over all the pixels in the image.

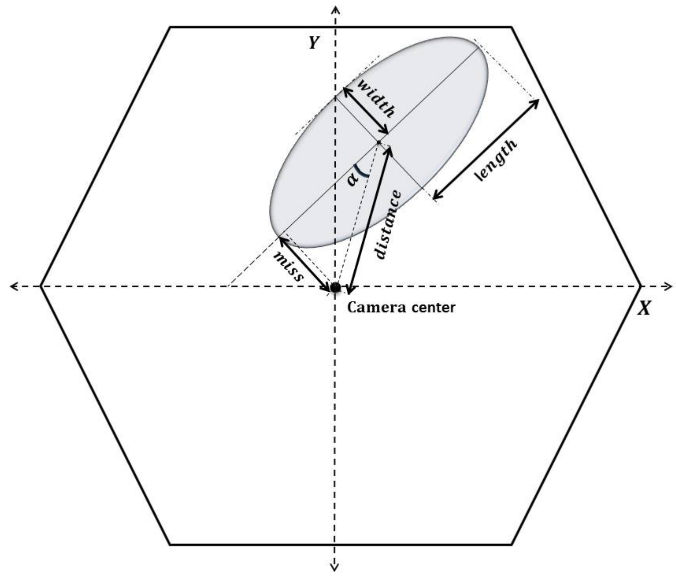

Figure 1: Definitions of some hillas parameters in the camera plane.

Image spreads are derived from the moments in (1):

$$\begin{aligned} &\sigma_x^2 = \langle x^2 \rangle - \langle x \rangle^2 \quad \sigma_{xy} = \langle xy \rangle - \langle x \rangle \langle y \rangle \\ &\sigma_y^2 = \langle y^2 \rangle - \langle y \rangle^2 \end{aligned} \tag{2}$$

Image parameters can be derived from the moments and the image spreads in (2):

$$\begin{aligned} length &= \sqrt{\frac{\sigma_x^2 + \sigma_y^2 + \sqrt{(\sigma_x^2 - \sigma_y^2)^2 + 4(\sigma_{xy})^2}}{2}} \\ width &= \sqrt{\frac{\sigma_x^2 + \sigma_y^2 - \sqrt{(\sigma_x^2 - \sigma_y^2)^2 + 4(\sigma_{xy})^2}}{2}} \\ distance &= \sqrt{\langle x \rangle^2 + \langle y \rangle^2} \\ \alpha &= \sin^{-1}\left(\frac{miss}{distance}\right) \end{aligned} \tag{3}$$

where, miss is the perpendicular distance between camera center and the major axis as shown in Fig. 1.

3 Methodology

3.1 Dataset

We are using the MAGIC Gamma Telescope Data Set in the UC Irvine Machine Learning Repository[15] which was created for the work in Ref [9]. These events were generated by a Monte Carlo program, Coriska with energies below $50GeV$[16]. The data set consist of two classes: gamma (signal) and hadron (background) with 12,332 gamma and 6688 hadron events. 10 chosen image parameters are used as features for classification.

3.2 Quantum Classification Algorithms

Quantum machine learning (QML) offers yet another potential direction to study data, find patterns and build classification models in a way that is very different from traditional classical machine leaning. QML offers new types of

Table 1: Image parameters

Classifiers	Description
Length	rms spread along the major axis of ellipse, a measure of longitudinal development of the shower .
Width	rms spread of light along the minor axis of ellipse.It is a measure of lateral development of the shower.
Size	10-log of total integrated light content in the image.
Conc 2	Ratio of sum of two brightest pixels over Size, a degree of light concentration.
Conc	Ratio of brightest pixel over Size.
Asym	Distance from the brightest pixel to center, projected onto major axis of the ellipse, a measure of asymmetry.
M3Long	Cube root of third moment along major axis of ellipse.
M3Trans	Cube root of third moment along minor axis of ellipse.
Alpha (α)	Angle between the image axis and the line joining the camera center and centroid of the ellipse.
Distance	Distance from the camera center to the centroid of the ellipse.

Table 2: Accuracy score for various models for the entire dataset

Model	**Accuracy**
QSVC	62
XGBClassifier	88
LGBMClassifier	88
RandomForestClassifier	88
ExtraTreesClassifier	88
BaggingClassifier	86
SVC	87
AdaBoostClassifier	84
DecisionTreeClassifier	81
KNeighborsClassifier	83
LabelSpreading	83
LabelPropagation	82
NuSVC	83
ExtraTreeClassifier	78
SGDClassifier	79
LogisticRegression	79
CalibratedClassifierCV	79
LinearSVC	79
LinearDiscriminantAnalysis	78
RidgeClassifier	78
RidgeClassifierCV	78
NearestCentroid	76
QuadraticDiscriminantAnalysis	78
BernoulliNB	76
Perceptron	76
PassiveAggressiveClassifier	70
GaussianNB	73
DummyClassifier	54

models that leverage quantum computers unique capabilities, for example, an exponentially higher-dimensional feature spaces to improve the accuracy of models. In this project we explore the data and the classification problem with once such QML algorithms called the Variational Quantum Classifier (VQC) using the IBM's recently launched tech stack Qiskit Machine Learning.VQC algorithm is a suitable candidate for any classification problem within QML in Noisy Intermediate-Scale Quantum(NISQ) era, although fault-tolerant quantum computers will likely not be available for few years. In the VQC, classical data is first mapped into a higher-dimensional feature space where the problem at hand becomes easier to solve. Then the VQC uses the traditional variational method to solve any problems on a quantum processor or simulator using classical optimizers to train a parameterized quantum circuit and provides a solution that cleanly separates the data. Hence, a VQC is also called a parameterized quantum circuit. The three major important components of VQC are the:

- Feature Map: Quantum algorithms require quantum data inputs for model training. Hence, a quantum embedding representing classical data as a Quantum state in Hilbert feature space with the help of gate parameters is used in the first layer and is called more generally a Feature Map.
- Variational Circuit: The next layer of circuit is constructed using ansatz and gates which gives us variational parameters(Θ) perform the learning and tuning the model.
- Optimizer: Optimizers are algorithms or methods used to change the attributes of variational circuits parameters(Θ) and iterations to reduce the losses.

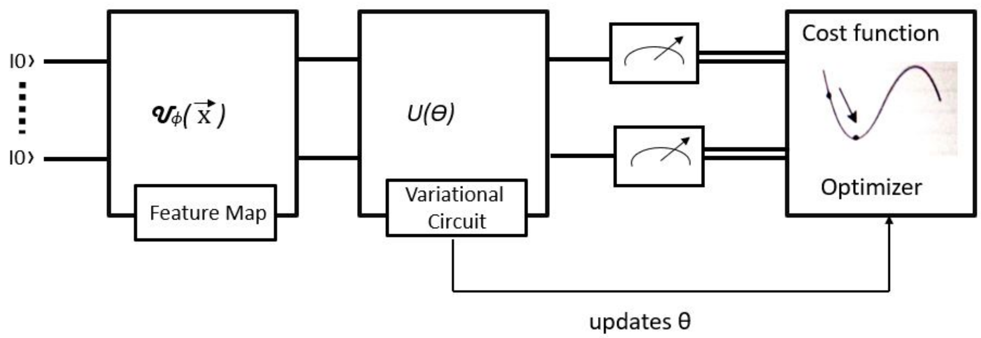

Figure 2: Pictorial components of the VQC classifier[17]

Second approach of QML is the Quantum Support Vector Classifier(QSVC) which consists of a quantum kernel for estimating the kernel function on a quantum computer and optimizes a classical SVM directly.The algorithm classifies a signal as gamma or hadron,by taking advantage of the high dimensionality of the quantum Hilbert space.The Quantum kernel plays a key role in separating the data with a hyperplane. The kernel $K(\vec{x}_i, \vec{x}_j)$ is estimated on a quantum computer for all pairs of training samples $\vec{x}_i, \vec{x}_j$ in the training dataset. After choosing the feature map, we apply this QSVC for the mapped data to create a hyperplane for separating gamma and hadrons.For getting quantum advantage the featuremap must be based on circuits which are hard to simulate using classical computers[11].

Table 3: Accuracy and Total time taken by QSVC using 1000 events for different feature maps.

	Train accuracy	**Test accuracy**	**Time Taken**
ZFeatureMap	100	57	0:10:40
PauliFeatureMap	100	66	1:22:19
ZZFeatureMap	100	63.5	1:27:20

Table 4: Accuracy and Total time taken by VQC using 1000 events for different feature maps.

	Train accuracy	**Test accuracy**	**Time Taken**
ZFeatureMap	63.5	62.5	3:45:13
PauliFeatureMap	61.125	56	4:20
ZZFeatureMap	55.5	51.5	4:50:45
RawFeatureVector	57.125	62	2:02:46

4 Hyper Parameter Tuning

We consider only 1000 events for Hyper Parameter tuning.The optimizer used here is COBYLA (Constrained optimization by linear approximation) and is executed on statevector simulator.COBYLA is much faster and will give better

Table 5: Accuracy and Total time taken by VQC using 1000 events for different variational circuits(ansatz).

	Train Accuracy	**Test Accuracy**	**Time Taken**
RealAmplitudes	55.875	56.5	2:00:00
EfficientSU2	62.875	56.5	2:45:23
PauliTwoDesign	60.75	56	1:45:24
NLocal	52.375	51.5	0:15:13
TwoLocal	63.5	62.5	0:15:13

Table 6: Hyper parameter search on QSVC with 1000 events

	kernel	**Test score**
0	(0.5, [Y], 1)	60.1786
1	(0.5, [Z], 1)	60.1786
2	(0.5, [YZ], 1)	61.0714
3	(0.5, [ZZ], 1)	61.0714
4	(1, [Y], 1)	60.3571
5	(1, [Z], 1)	60.3571
6	(1, [YZ], 1)	61.0714
7	(1, [ZZ], 1)	61.0714
8	(2, [Y], 1)	60.3571
9	(2, [Z], 1)	60.3571
10	(2, [YZ], 1)	61.0714
11	(2, [ZZ], 1)	61.0714
12	(0.5, [Y], 1)	55.8929
13	(0.5, [Z], 1)	55.8929
14	(0.5, [YZ], 1)	61.0714
15	(0.5, [ZZ], 1)	61.0714
16	(1, [Y], 1)	60.1786
17	(1, [Z], 1)	60.1786
18	(1, [YZ], 1)	61.2500
19	(1, [ZZ], 1)	61.0714
20	(2, [Y], 1)	60.1786
21	(2, [Z], 1)	60.1786
22	(2, [YZ], 1)	60.8929
23	(2, [ZZ], 1)	61.0714

performance than SPSA(Simultaneous perturbation stochastic approximation) optimizer. The number of optimization iterations is fixed as 100. If we further increase the number of optimization iterations the time taken will increase and there is no significant improvement in the accuracy.

Large number of features requires high computational resources and may even lead to overfitting.Feature map is used for reducing the dimensionality of the data. For QSVC algorithm PauliFeatureMap has the highest testing accuracy and ZFeatureMap has the shortest time taken. The training accuracy is 1 for all Feature maps. We have chosen Pauli Feature map for QSVC Algorithm. The Hyper-parameter search algorithm will automatically optimize the hyper-parameters of the quantum machine learning model. Hyper-parameters are the parameters which are not updated during the learning and are used to configure the algorithm. Hyper parameter search using 1000 events and Pauli Feature Maps with Y,Z,YZ,ZZ configurations will give rise to 24 possible fittings. The best kernel is (1,['YZ'],1) with a test score of 0.61.Entangler map assosciated with a featuremap is used to specify the entanglement of qubits.Here we used linear entanglement so that each qubit j is entangled with the qubit $j + 1$.The number of repitions of the featuremap is 3.

Whereas for VQC Algorithm RawFeatureVector and ZFeatureMap performs.We found that TwoLocal circuit has the highest accuracy and the time taken is lesser compared to others.Hence for tuning the feature map we are choosing Two Local as the variational circuit for VQC.

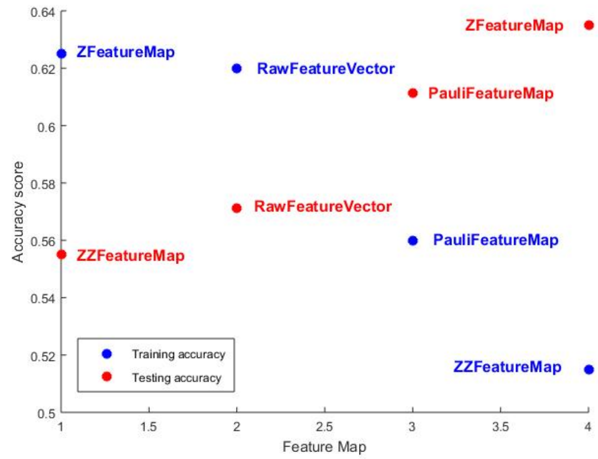

Figure 3: Accuracy score of various feature maps for VQC with 1000 events on IBM Statevector Simulator and using COBYLA optimizer with 100 iterations.

Table 7: Hyperparameter Search on QSVC for different sets of events

Events	**Test accuracy**	**Train accuracy**
100	60.00	100
200	60	96.43
300	60	100
500	58	91.43
700	69.29	100
900	69.29	100
1000	63.50	100

5 Quality Parameters

The gamma acceptance efficiency $\epsilon_\gamma = \frac{N_\gamma}{S}$ and the hadron acceptance efficiency is given by $\epsilon_h = \frac{N_h}{B}$, where S and B are the total number of gamma (signals) and hadron(background) events respectively. N_γ and N_h are the number of gamma and hadronic events which will survive the discrimination process. The Quality factor is defined as

$$Q = \frac{\epsilon_\gamma}{\sqrt{\epsilon_h}} \tag{4}$$

and the statistical significance is given by,

$$\sigma = N_\gamma / \sqrt{2N_h + N_\gamma} \tag{5}$$

We are interested only in the value of Q obtained at $\epsilon_\gamma = 0.5$.

Receiver Operating Characteristics(ROC) is the plot of signal efficiency (ϵ_γ) vs. background rejection(ϵ_h). The Area under the Receiver Operating Characteristics (AUC) is used as a metric to compare the performance of different classifiers. $liacc$ can be calculated by interpolating the ROC curve and taking the mean of ϵ_γ at the points where ϵ_h= 0.01,0.02 and 0.05. $hiacc$ is the mean of ϵ_γ at the points $\epsilon_h = 0.1$ and $\epsilon_h = 0.2$ [9].

Table 8: Accuracy and Area under the ROC curves(AUC) for VQC on 5k events

	Test accuracy	**Train accuracy**
VQC (statevector simulator)	64.9	63.1
VQC (qasm simulator)	62.2	60
QSVC(statevector simulator)	64.1	100
XG Boost	88.4	97.9

Table 9: Comparison of Accuracy obtained from QSVC, Classical SVC and XGBoost

Events	**QSVC**	**SVC**	**XGBoost**
100	60	75	70
200	67.5	80	75
300	63.3	83.3	78.33
500	62	87	74
700	65.7	83.57	77.14
900	68.3	88.3	77.22
1000	65.5	86	77.5

6 Clustering Architecture for VQC

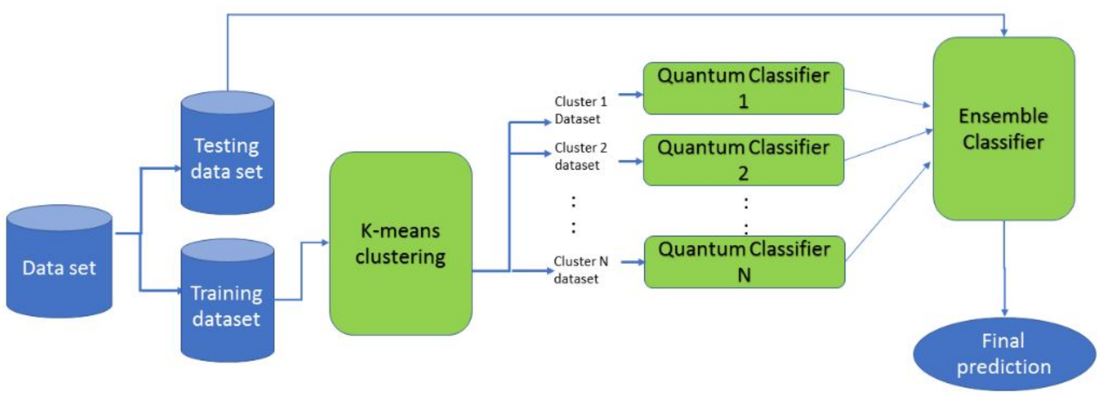

Figure 4: Clustering Architecture for QSVC and SVC

Quantum Classifier Algorithms can be very slow for certain large datasets. Using the architecture shown in Fig. 4. it is possible to use Quantum Classifier algorithms on large datasets efficiently. Here the dataset is split into training and testing datasets with 8:2 ratio. Training dataset is clustered to an optimal number of small clusters. We have used k-means clustering, however one can also go for Quantum Clustering. We found that clustering prior to classification is useful to enhance the overall performance. The dimensionality of the data is reduced using PCA before clustering. Each cluster dataset is then independently classified by the QSVC/SVC model.An ensemble classifier is implemented using the voting classifier in scikit learn. The ensemble classifier will combine all the models and final predictions is made on the testing data.

Table 10: Accuracy and AUC of QSVC and XGBoost Algorithm for 10k events in statevector simulator

	AUC	**Test accuracy**	**Train accuracy**
QSVC (statevector simulator)	51.65	62.75	100
XG Boost	94.02	88.55	95.58

Table 11: Test Accuracy for QSVC and Classical SVC after clustering

Cluster	**Size**	**QSVC**	**Classical SVC**
Cluster 1	500	71	78.25
Cluster 2	500	77	82.75
Cluster 3	500	70	82.00

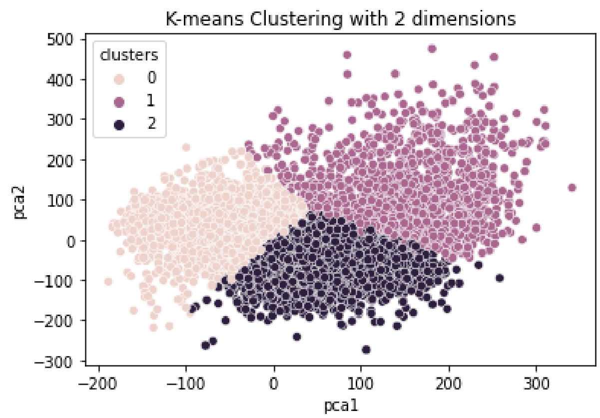

Figure 5: k-means clustering(3 clusters)

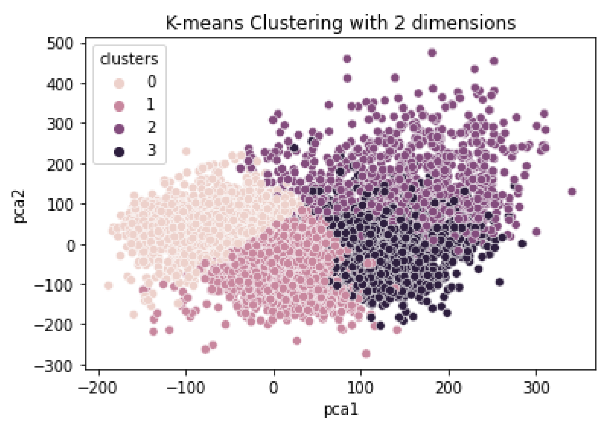

Figure 6: k-means clustering(4 clusters)

7 Conclusion and future scope

In this study, we have successfully employed Quantum Machine Learning Algorithms for gamma hadron separation in Imaging Atmospheric Cherenkov Telescopes on gate-model quantum computer simulators. We have used 10 qubits and up to 19k events in IBM Quantum Framework. We have successfully run VQC for 5k events on both statevector simulator and qasm simulator. The VQC algorithm is very slow compared to QSVC. QSVC algorithm achieves a classification performance similar to the performances of the classical machine learning methods. For entire dataset QSVC algorithm has achieved a testing accuracy score of 62.02. We have chosen the best possible combinations of feature maps, ansatz/kernels and optimizers using Hyper parameter tuning on 1000events. QSVC Algorithm is

Table 12: Test Accuracy of QSVC and Classical SVC after clustering

Cluster	**Size**	**QSVC**	**Classical SVC**
Cluster 1	500	64	84.00
Cluster 2	500	70	77.5
Cluster 3	500	96	95
Cluster 4	500	69	80.50

Table 13: Test Accuracy of Ensemble Classifier

Number of clusters	**QSVC**	**Classical SVC**
3	65.1	80.4
4	65.5	81.1

run on 10k events on statevector simulator and the Quality parameters obtained were compared to that of XGBoost Algorithm. However classical simulators cannot efficiently simulate QSVC algorithm as it is hard to estimate the kernels in quantum feature space using classical computers.Using the proposed Clustering architecture Quantum Classifiers can be implemented efficiently on larger datasets.

References

[1] Idan Shilon, Manuel Kraus, Matthias Büchele, Kathrin Egberts, Tobias Fischer, Tim Lukas Holch, Thomas Lohse, Ullrich Schwanke, Constantin Steppa, and Stefan Funk. Application of deep learning methods to analysis of imaging atmospheric cherenkov telescopes data. *Astroparticle Physics*, 105:44–53, 2019.

[2] John David Jackson. Classical electrodynamics, 1999.

[3] AM Hillas. Differences between gamma-ray and hadronic showers. *TeV Gamma-Ray Astrophysics*, pages 17–30, 1996.

[4] Michael F Cawley, DJ Fegan, K Harris, AM Hillas, PW Kwok, RC Lamb, MJ Lang, DA Lewis, D Macomb, PT Reynolds, et al. A high resolution imaging detector for tev gamma-ray astronomy. *Experimental Astronomy*, 1(3):173–193, 1990.

[5] David J Fegan. /hadron separation at tev energies. *Journal of Physics G: Nuclear and Particle Physics*, 23(9):1013, 1997.

[6] Tobias Voigt, Roland Fried, Michael Backes, and Wolfgang Rhode. Gamma-hadron-separation in the magic experiment. In *Data Analysis, Machine Learning and Knowledge Discovery*, pages 115–124. Springer, 2014.

[7] J Albert, E Aliu, H Anderhub, P Antoranz, A Armada, M Asensio, C Baixeras, JA Barrio, H Bartko, D Bastieri, et al. Implementation of the random forest method for the imaging atmospheric cherenkov telescope magic. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 588(3):424–432, 2008.

[8] EB Postnikov, AP Kryukov, SP Polyakov, DA Shipilov, and DP Zhurov. Gamma/hadron separation in imaging air cherenkov telescopes using deep learning libraries tensorflow and pytorch. In *Journal of Physics: Conference Series*, volume 1181, page 012048. IOP Publishing, 2019.

[9] RK Bock, A Chilingarian, M Gaug, F Hakl, Th Hengstebeck, M Jiřina, J Klaschka, E Kotrč, P Savickỳ, S Towers, et al. Methods for multidimensional event classification: a case study using images from a cherenkov gamma-ray telescope. *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment*, 516(2-3):511–528, 2004.

[10] Richard P Feynman. Simulating physics with computers. In *Feynman and computation*, pages 133–153. CRC Press, 2018.

[11] Vojtěch Havlíček, Antonio D Córcoles, Kristan Temme, Aram W Harrow, Abhinav Kandala, Jerry M Chow, and Jay M Gambetta. Supervised learning with quantum-enhanced feature spaces. *Nature*, 567(7747):209–212, 2019.

[12] Patrick Rebentrost, Masoud Mohseni, and Seth Lloyd. Quantum support vector machine for big data classification. *Physical review letters*, 113(13):130503, 2014.

[13] A Michael Hillas. Cerenkov light images of eas produced by primary gamma. In *19th International Cosmic Ray Conference (ICRC19), Volume 3*, volume 3, 1985.

[14] AM Hillas. The sensitivity of cerenkov radiation pulses to the longitudinal development of cosmic-ray showers. *Journal of Physics G: Nuclear Physics*, 8(10):1475, 1982.

[15] Dheeru Dua and Casey Graff. UCI machine learning repository, 2017.

[16] Dieter Heck, J Knapp, JN Capdevielle, G Schatz, T Thouw, et al. Corsika: A monte carlo code to simulate extensive air showers. *Report fzka*, 6019(11), 1998.

[17] Aaron Baughman, Daniel Bohm, Micah Forster, Eduardo Morales, Jeff Powell, Shaun McPartlin, Raja Hebbar, and Kavitha Yogaraj. Large scale diverse combinatorial optimization: Espn fantasy football player trades. *arXiv preprint arXiv:2111.02859*, 2021.