Received: Added at production | Revised: Added at production | Accepted: Added at production
DOI: xxx/xxxx

ORIGINAL RESEARCH

Exploring Quantum Single-Qubit Method for Pulsar Classification in Astronomical Data

Jayesh V. Hire[1] | Vaidehi Gawande[2] | Sagar Dhande[1]

[1] TrevasQ Private Ltd, Pune,Maharashtra, India
[2]Creed & Bear Network LLC, Dubai, UAE

Correspondence
Jayesh V. Hire
Email: hire.jayesh123@gmail.com

Abstract

Pulsar stars, characterized by their highly magnetized and rapidly rotating compact nature, as well as their emission of radiation beams from magnetic poles, play a crucial role in astrophysics, particularly in the study of gravitational waves and general relativity. This paper investigates the application of machine learning techniques for the classifications of pulsars using radio wave emission data from the HTRU-2 dataset, which contains an extensive collection of candidate samples. The study explores two quantum-assisted methodologies: a 9-qubit Quantum Approximate Optimization Algorithm (QAOA) inspired encoding and an innovative single-qubit Quantum Asymptotically Universal Multi-feature (QAUM) encoding within a quantum neural network framework. The performance evaluation encompasses a range of metrics, including accuracy, precision, recall, and specificity, which are analyzed across various optimizer configurations and learning rates. Notably, the QAUM approach consistently demonstrates superior sensitivity and reduced error rates compared to QAOA, particularly in specific learning rate settings. These findings underscore the transformative potential of quantum computing in advancing pulsar detection methodologies, highlighting its efficacy and implications for astrophysical data analysis and the broader field of quantum-enhanced scientific exploration.

KEYWORDS
quantum computing, quantum computing techniques

1 | INTRODUCTION

In the field of astronomy, it is essential to gather and classify large volumes of observational data related to different celestial objects. This research is concerned with pulsars, which are a specific type of neutron star that originates from the collapse of a massive star towards the end of its lifecycle. The collapse occurs due to gravity but is halted by neutron degeneracy pressure, leading to the formation of a dense remnant with a small radius and fast rotation, often ranging from seconds to milliseconds [Maoz (2016)][de Groot (1977)]. Pulsars are distinguished by their powerful magnetic fields that emit particle beams from their poles. These beams produce periodic pulses when they intersect with Earth and are typically identified using radio telescopes.

Identifying pulsars is crucial for distinguishing them from main sequence stars and other neutron stars, especially as binary systems of pulsars can produce gravitational waves [Hulse and Taylor (1975)][Stairs (2003)]. The exact rotational periods of pulsars are valuable in detecting gravitational waves passing between them and Earth, as these waves cause irregularities in the regular pulses [McLaughlin (2013)][Verbiest et al. (2016)]. This capability significantly enhances our comprehension of gravitational waves.

The rapid progress in observational technology has resulted in a substantial increase in astronomical data, posing challenges to traditional software's capacity to handle and process these extensive datasets. The upcoming Square Kilometer Array, predicted to generate exabytes (10^{18} bytes) of data, highlights the pressing requirement for real-time data analysis and efficient categorization methods [An (2019)]. While current techniques are adequate, future data volumes will surpass their capabilities. A key scientific objective of the SKA is to explore and analyze pulsars; the world's largest radio telescope aims to conduct extensive research on the pulsar population with an emphasis on addressing fundamental questions in modern physics such as identifying and charting gravitational waves within our galaxy [Keane et al. (2014)]. By measuring the timing of radio pulses from numerous millisecond pulsars [Hobbs et al. (2010)][Janssen et al. (2014)], it can detect gravitational waves

Abbreviations: QAUM, quantum asymptotically universal multi-feature; QAOA, quantum approximate optimization algorithm.

by observing disruptions in these pulses which introduces a new detection method that complements ground-based experiments like LIGO[] Abbott et al. (2016)].

To accomplish these goals, the SKA needs to locate and survey thousands of previously undiscovered pulsars by distinguishing their regular signals from other sources of interference at radio frequencies. Therefore, there is an increasing emphasis on developing advanced classification algorithms [Lyon et al. (2016)]. In this scenario, extensive research has been conducted on using machine learning as a potential solution to this issue. While classical machine learning techniques have been utilized for pulsar classification [Beniwal et al. (2021)][Xiao et al. (2020)][Tariq et al. (2022)], they encounter challenges with growing data volumes [Zhou et al. (2017)]. To overcome these limitations, a fresh approach may be required such as quantum computing – specifically quantum machine learning – which merges the computational capabilities of quantum computers with machine learning and presents an alternative perspective in classifying pulsars. This investigation examines the suitability of employing quantum computers in classifying pulsars. This is accomplished by assessing the performance of two different quantum methods outlined in this study, both using the HTRU-2 dataset. Our objective is to evaluate and enhance their efficiency in categorizing pulsars by adjusting hyperparameters such as optimizers and learning rates.

The accuracies and losses in this scenario have been enhanced by fine-tuning hyperparameters while maintaining the same depth of circuits. This research effectively identifies pulsars using the HTRU-2 dataset collected from real-world observations. The study examines and compares the performance of two distinct quantum-based approaches: a standard QAOA encoding (in our model, a 9-qubit architecture) and a novel single-qubit quantum encoding termed the Quantum Asymptotically Universal Multi-feature (QAUM) encoding, within a quantum neural network framework [Kordzanganeh et al. (2021)][Farhi et al. (2000)]. The evaluation involves rigorous comparison across various performance metrics, utilizing different optimizers and learning rates to thoroughly assess the efficacy of these models.

The subsequent sections are organized as follows: Section 2 explores binary classification as a supervised machine learning task, providing detailed insights into the utilization of various optimizers. Section 3 outlines the structure of the HTRU-2 dataset, emphasizing its integration into the model implementation. Section 4 undertakes an extensive comparative analysis between the two methodologies, complemented by visual aids. Finally, Section 5 offers a concise concluding discussion.

2 | THEORETICAL BACKGROUND

2.1 | Binary Classification & Evaluation Metrics

Binary classification is a supervised machine learning problem in which each instance in a dataset is allocated to one of two separate categories. This scenario aims to ascertain whether an instance should be categorized as a pulsar or non-pulsar. Below is the precise definition of binary classification [Schuld and Petruccione (2021)].

$$f_\theta : X \rightarrow Y, \quad f_\theta(\vec{x}) = y, \quad \vec{x} \in X \text{ and } y \in Y \tag{1}$$

where, X and Y denote the input and output domains, respectively. Each input vector x comprises n features, while the corresponding output y belongs to a set $\{0, 1\}$, where 1 denotes a pulsar and 0 represents a non-pulsar. The optimal model $f_{\hat{\theta}}$ is chosen from a collection of models $\{f_\theta\}$, with θ representing the parameters optimized during training.

During the training process, a portion of data, the training set (X_{train} and Y_{train}), is utilized to fine-tune or customize a model from a specific category. This model's objective is to accurately forecast labels for the testing set (X_{test} and Y_{test}). By comparing these forecasts with the actual labels, a confusion matrix a 2×2 array captures the prediction results. Confusion matrices visually depict how well predictions align with real outcomes in new samples. Once an effective model has been developed through training, it can be used to predict labels for unknown data without any prior knowledge of their labels.

Various metrics are chosen to assess how well a model performs in making predictions. Accuracy calculates the percentage of correctly classified samples compared to the known labels. Recall, also known as sensitivity, measures the proportion of positive cases correctly identified as positive. Similarly, specificity complements recall but focuses on negative cases. Precision evaluates the proportion of positively identified samples that belong to the positive class. The higher the recall and precision, the more effective the model is at predicting the positive class [Tharwat (2020)][Powers (2020)].

2.2 | Optimizers

Optimizers play a crucial role in training machine learning models by modifying model parameters to reduce the loss function. This study explores four commonly used optimizers: Adam, Adagrad, Gradient Descent, and Nesterov Momentum. Each optimizer has distinct strengths and weaknesses, and understanding their properties can significantly impact both the training process and the model's effectiveness [PennyLane Documentation (2024)].

2.2.1 | Adam Optimizer

Adam, which stands for Adaptive Moment Estimation, is an adaptive optimization method that combines elements from AdaGrad and RMSProp. It computes individual adaptive learning rates for different parameters by tracking exponentially decaying averages of past gradients and squared gradients. This allows Adam to adjust the learning rate during training, leading to faster convergence and enhanced performance, especially in situations with sparse gradients or noisy data [Kingma and Ba (2014)][Ruder (2016)].

Adam updates the parameters using the following formulas:

$$x^{(t+1)} = x^{(t)} - \eta^{(t+1)} \frac{a^{(t+1)}}{\sqrt{b^{(t+1)}} + \epsilon} \tag{2}$$

The update rules for the two moments are given by:

$$a^{(t+1)} = \beta_1 a^{(t)} + (1 - \beta_1)\nabla f(x^{(t)}) \tag{3}$$

$$b^{(t+1)} = \beta_2 b^{(t)} + (1 - \beta_2)(\nabla f(x^{(t)}))^{\odot 2} \tag{4}$$

$$\eta^{(t+1)} = \eta \frac{\sqrt{(1 - \beta_2^{t+1})}}{(1 - \beta_1^{t+1})} \tag{5}$$

Here, the expression $(\nabla f(x^{(t-1)}))^{\odot 2}$ denotes the element-wise squaring operation, signifying that each element in the gradient is multiplied by itself. The hyperparameters β_1 and β_2 may also differ at each iteration. At first, both the initial and subsequent moments are initialized to zero. The term ϵ is introduced to avoid division by zero.

2.2.2 | Adagrad Optimizer

Adagrad, also called the Adaptive Gradient Algorithm, is a method used to adapt the learning rate in optimization. It customizes the learning rate for each parameter based on their past gradients. This approach increases the learning rate for parameters that are updated infrequently and decreases it for those updated frequently. Adagrad is particularly useful for datasets with sparse features where certain features hold more importance than others. However, one potential drawback of Adagrad is that it may experience diminishing learning rates over long training periods, causing a decrease in the speed of learning [Duchi et al. (2011)][Ruder (2016)].

Adagrad updates the parameters using the following formula for each parameter x_i:

$$x_i^{(t+1)} = x_i^{(t)} - \eta_i^{(t+1)} \partial_{w_i} f(x^{(t)}) \tag{6}$$

The learning rate in step t is given by:

$$\eta_i^{(t+1)} = \frac{\eta_{\text{init}}}{\sqrt{a_i^{(t+1)} + \epsilon}} \tag{7}$$

Where

$$a_i^{(t+1)} = \sum_{k=1}^{t} (\partial_{x_i} f(x^{(k)}))^2 \tag{8}$$

The offset ϵ avoids division by zero. η is the step size, a user-defined parameter.

2.2.3 | Gradient Descent Optimizer

The Gradient Descent method is a key optimization technique used to reduce the loss function by gradually moving in the direction of the steepest descent on the surface of the loss. It updates the parameters of the model by subtracting a fraction of the gradient of the loss function with respect to those parameters. Although it is simple and can be implemented directly, Gradient Descent may demonstrate slow convergence, especially for high-dimensional or non-convex optimization problems [Ruder (2016)].

Gradient Descent updates the parameters using the following formula:

$$x^{(t+1)} = x^{(t)} - \eta \nabla f(x^{(t)}) \tag{9}$$

Where η is a user-defined hyperparameter corresponding to the step size.

2.2.4 | Nesterov Momentum Optimizer

Nesterov Momentum is an extension of the traditional momentum optimization algorithm. Instead of simply using past gradients, it calculates a decreasing average and uses this to update parameters. Unlike standard momentum, Nesterov Momentum looks ahead to assess the gradient at a future point, allowing it to anticipate the direction of the next update. This proactive adjustment speeds up convergence and improves performance, especially in situations with noisy gradients or significant curvature [Ruder (2016)].

Nesterov Momentum updates the parameters using the following formula:

$$a^{(t+1)} = ma^{(t)} + \eta \nabla f(x^{(t)} - ma^{(t)}) \tag{10}$$

Where η is the step size and m is the momentum.

3 | METHODOLOGY

3.1 | The Dataset

The dataset used for this comparative analysis, HTRU-2, has been preprocessed specifically for machine learning applications [Lyon (2017)]. It contains 8 features per candidate sample with a binary classification label of 0 or 1. A value of 1 indicates a positive class (a pulsar). These features are grouped into statistical elements representing both the integrated field and DM-SNR (Dispersion Measure-Signal to Weight Ratio) curves of each candidate. These statistical elements include mean, standard deviation, excess kurtosis, and skewness. Furthermore, a correlation matrix was generated to investigate the relationships among these variables, highlighting their interdependencies and their influence on the classification process, Figure 1.

Out of the total 17,898 candidates in the dataset, only 1,639 have been confirmed as pulsar samples while the remaining 16,259 are spurious samples caused by noise or radio frequency interference. For further details on the integrated field and DM-SNR curves' significance in relation to pulsar classification please refer to the sources [Lyon (2016)]. All features were normalized within the range from 0 to π radians to match the rotational encoding strategy selected [Kordzanganeh et al. (2021)].

The dataset was divided using a stratified sampling strategy, preserving an 80:20 ratio for the training and testing sets, respectively. To

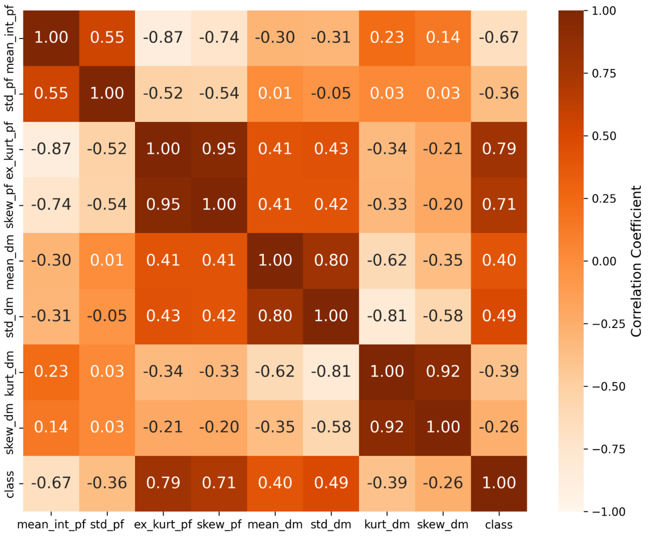

FIGURE 1 A correlation matrix illustrates the correlation coefficients among all feature pairs in the HTRU-2 dataset.

ensure balanced representation within the training set, an equal number of samples from both the positive and negative class candidates were randomly selected. The remaining samples were then designated as the testing set. This methodological approach enables flexibility in adjusting the sizes of the training and testing sets based on specific requirements, thereby ensuring robustness in evaluating the model's performance across different data partitions.

3.2 | Model Implementation

The quantum model makes use of the Quantum Universal Multi-feature Encoding known as QAUM, which is based on a previous study [Kordzanganeh et al. (2021)] and expands upon single-qubit models examined in [Schuld et al. (2021)]. Furthermore, the model incorporates the Quantum Approximate Optimization Algorithm encoding. These two distinct approaches are utilized to predict pulsars with different sets of hyperparameters, including various optimizers and learning rates. The training process involved the application of a cross-entropy loss function. The model was trained for 150 epochs.

The QAOA approach, which utilizes Y rotations, was directly brought in from the PennyLane templates to train the network. This configuration, applied on 9 qubits, included 18 adjustable parameters for each iteration.

The depth, denoted as L, is set to 2 for QAUM and 3 for QAOA. This corresponds to the number of data reuploading layers in the respective circuits. As the number of training parameters for each model increases differently, the models' parameter counts are most similar when L=3 repetitions for QAOA (54 parameters) and L=2 repetitions for QAUM (51 parameters).

In our comparative analysis of the methods described in the paper [Kordzanganeh et al. (2021)], we assess the model using various optimization algorithms such as Adam, Adagrad, Gradient Descent, and Nesterov Momentum. This evaluation includes two different learning rates: 0.1 and 0.01.

4 | RESULT

The referenced circuits were simulated utilizing PennyLane's default.qubit device, with gradients calculated using the simulator's backpropagation technique. It should be noted that the PennyLane framework provides a lightning.qubit device that is optimized for computational efficiency compared to the default.qubit device. However, due to compatibility issues with backpropagation techniques, the default.qubit device was utilized for the simulations performed in this study [PennyLane (2024). The classical device used to simulate all the noise-free operations has the following specifications: an Acer Aspire A515-52G laptop with an Intel® Core™ i5-8265U CPU @ 1.60GHz × 8, 8 GB RAM, and Ubuntu 22.04.2 LTS.

The confusion matrices presented in Figure 2 , which were generated using 360 training samples and 120 testing samples, demonstrate that both the QAUM and QAOA methods effectively differentiate between pulsars and non-pulsars. The QAUM model, with a depth of L=2, demonstrates a modest performance advantage over the QAOA model, which has a depth of L=3. The QAUM model exhibits higher sensitivity compared to the QAOA model, as evidenced by its ability to correctly identify a greater number of pulsars. This is reflected in the lower false negative rate, which indicates that the QAUM model is more effective at detecting pulsars. Conversely, the QAOA model demonstrates a higher false negative rate, suggesting it is less effective at identifying pulsars.

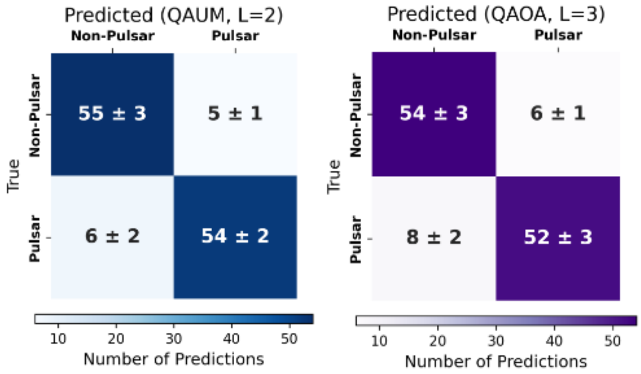

FIGURE 2 The confusion matrices for experiments conducted with 360 training samples and 120 testing samples illustrate the performance of the QAUM (Blue) and QAOA (Purple) techniques. Each entry in the confusion matrices is the average of five distinct runs and the uncertainty is the standard error.

TABLE 1 Optimizer Comparison for QAUM & QAOA (LR 0.1)

Metric	QAUM (L= 2)				QAOA (L= 3)			
	Adam	Adagrad	GD	NM	Adam	Adagrad	GD	NM
Min. Training Loss	0.1935	0.2952	0.2375	0.1846	0.2673	0.2634	0.3185	0.2647
Training Accuracy	0.9444	0.9306	0.9278	0.9417	0.9139	0.9111	0.8694	0.9056
Accuracy	0.9067	0.8883	0.8972	0.905	0.885	0.89	0.8517	0.88
Precision	0.9165	0.8897	0.9119	0.9241	0.9018	0.8588	0.9021	0.9036
Recall	0.8991	0.8926	0.8767	0.8861	0.8645	0.9166	0.7982	0.8562
Specificity	0.9153	0.8885	0.9258	0.9281	0.9058	0.9161	0.908	0.8979

Concerning false positive rates, both models demonstrate strong performance, yet the QAUM model exhibits a slightly superior outcome. The confusion matrices suggest that the QAUM model has a somewhat lower rate of non-pulsars being incorrectly classified as pulsars in comparison to the QAOA model. The error margins for both models are comparable, signifying consistent and reliable outcomes across various trials. This consistency implies that both quantum-based methods are robust for the task of pulsar classification.

The Table 1, presents a detailed comparative analysis of the performance metrics for the QAUM and QAOA models across various optimization algorithms, including Adam, Adagrad, Gradient Descent (GM), and Nesterov Momentum (NM). All models were trained using a learning rate of 0.1. The comparative evaluation is based on four classification metrics: accuracy, precision, recall, and specificity.

The QAUM model demonstrated superior performance in various key metrics compared to the QAOA model. Notably, QAUM achieved the lowest minimum training loss of 0.1935 using the Adam optimization algorithm, indicating its efficient convergence during the training process. In contrast, the QAOA model's best performance, observed with the NM optimization algorithm, resulted in a higher minimum training loss of 0.2647. Furthermore, QAUM attained the highest training accuracy of 0.9444 using the Adam optimizer, surpassing QAOA's best training accuracy of 0.9417 with NM. This advantage in training accuracy suggests that the QAUM model was more effective in the learning process.

When evaluating overall performance on the test set, the QAUM model exhibited superior accuracy compared to the QAOA model across all optimization algorithms. Specifically, QAUM achieved the highest accuracy of 0.9067 using the Adam optimization method, surpassing QAOA's best accuracy of 0.885 with Adagrad. This suggests that QAUM demonstrates a stronger ability to generalize and achieve accurate classifications. In terms of precision, QAUM attained the highest precision of 0.9165 using Adam, indicating a greater capacity to minimize false positive identifications. Additionally, QAUM showcased higher recall, with a best of 0.8991 using Adam, reflecting its effectiveness in detecting true pulsars. Regarding specificity, which measures the ability to identify non-pulsars correctly, both models performed well, but QAUM's highest specificity of 0.9281 using the NM method slightly outperformed QAOA's best of 0.9161 with Adagrad. Collectively, these results highlight the robustness and reliability of the QAUM model, making it a more favorable choice for pulsar classification tasks where high accuracy and minimal error rates are of paramount importance.

Analogously, experiments with a learning rate of 0.01 were carried out, and the findings are summarized in Table 2. QAUM exhibited a superior minimum training loss of 0.2027 when utilizing the Adam optimizer, exceeding QAOA's best result of 0.2763 with the same optimizer. Furthermore, QAUM achieved a higher training accuracy of 0.9389 with Adam and a consistent precision of 0.9111 with Adagrad. In contrast, QAOA's performance exhibited more variability, with its highest precision being 0.9279 when employing the NM algorithm.

Regarding classification accuracy, the QAOA algorithm achieved its best result of 0.8883 using the NM optimization method, slightly outperforming the highest accuracy of 0.8767 attained by QAUM using the Adagrad optimizer. However, QAUM demonstrated superior recall performance, reaching 0.8700 with NM, indicating better detection of true positive instances compared to QAOA's best recall of 0.8627 using the Adam optimizer. Conversely, QAOA's top specificity of 0.9389 with NM was slightly higher than QAUM's specificity of 0.9114 with Adagrad. These findings highlight QAUM's stable and consistent performance across various evaluation metrics, while QAOA exhibits occasional exceptional performance in specific areas, particularly precision and specificity. To visually illustrate these contrasting results, histogram plots (Figure 3) of the metrics are included, providing a clear comparative analysis of each model's performance across the different optimization algorithms.

5 | CONCLUSION

Pulsars are among the most extensively researched natural occurrences in radio astronomy. The findings of this investigation highlight the significant potential of quantum-assisted machine learning algorithms for the detection of pulsar stars. Through a comparative analysis of a standard 9-qubit Quantum Approximate Optimization Algorithm (QAOA) inspired encoding and a novel single-qubit Quantum Asymptotically

TABLE 2 Optimizer Comparison for QAUM & QAOA (LR 0.01)

Metric	QAUM (L= 2)				QAOA (L= 3)			
	Adam	Adagrad	GD	NM	Adam	Adagrad	GD	NM
Min. Training Loss	0.2027	0.3183	0.2623	0.2359	0.2763	0.4437	0.4771	0.3253
Training Accuracy	0.9389	0.8944	0.9028	0.9167	0.9083	0.8861	0.8611	0.8917
Accuracy	0.8617	0.8767	0.8167	0.8883	0.8833	0.8417	0.6683	0.8583
Precision	0.8697	0.9111	0.8284	0.9059	0.8947	0.9089	0.7059	0.9279
Recall	0.8540	0.8429	0.7941	0.8700	0.8627	0.7504	0.6695	0.7760
Specificity	0.8762	0.9114	0.8386	0.9111	0.8997	0.9272	0.6622	0.9389

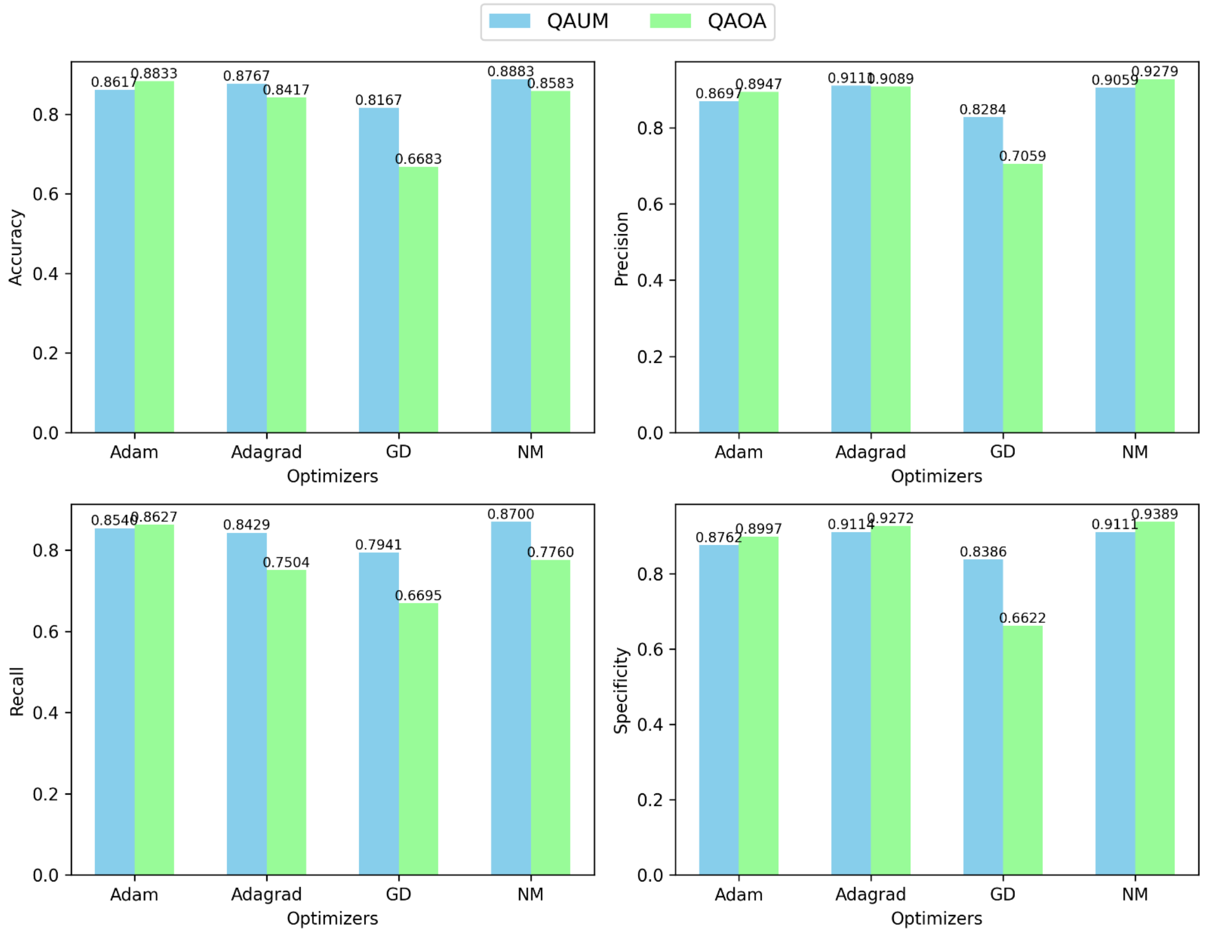

FIGURE 3 Comparison of metrics for different optimizers (QAUM vs QAOA) with learning rate 0.01

Universal Multi-feature (QAUM) encoding within a quantum neural network [Kordzanganeh et al. (2021)] framework, notable performance variations were observed.

The QAUM framework demonstrated consistently superior performance compared to the QAOA approach across multiple evaluation metrics. For example, at a learning rate of 0.1, the QAUM method achieved an accuracy rate of 90.67%, outpacing the QAOA's 88.50% accuracy when using the Adam optimizer. Furthermore, the QAUM approach exhibited higher precision and recall, with precision at 92.41% versus QAOA's 90.36%, and recall at 88.61% compared to QAOA's 85.62%. Additionally, the QAUM framework exhibited greater specificity, scoring 92.81% compared to QAOA's 89.79%.

The comprehensive evaluation of various optimization algorithms and learning rate settings further accentuated the benefits of the

(QAUM), demonstrating enhanced sensitivity and reduced error margins. For instance, the minimum training loss attained by QAUM employing the NM optimization method was 0.1846, QAOA it was 0.2647. These significant improvements suggest that QAUM's encoding approach more effectively addresses the complexities inherent to pulsar data analysis.

The presented findings highlight the substantial potential of quantum computing in advancing the analysis of astrophysical data. The demonstrated superior performance of the QAUM approach suggests its promise in enhancing pulsar detection methods, thereby enabling further research in gravitational waves and general relativity. Future investigations should focus on exploring the scalability of QAUM encoding when applied to larger datasets and noisier environments, in order to fully realize its capabilities for real-world applications. This study establishes a foundation for integrating quantum machine learning techniques into mainstream astrophysical research, offering more accurate and efficient data processing solutions.

AUTHOR CONTRIBUTIONS

Jayesh Hire: Data curation; formal analysis; investigation; methodology; resources; validation; visualisation; writing– original draft; writing – review & editing. **Vaidehi Gawande:** Conceptualisation; formal analysis; investigation; supervision; validation; writing – review & editing. **Sagar Dhande:** Investigation; resources; supervision; validation; writing – review.

6 | DATA AVAILABILITY STATEMENT

The data set analyzed during the current study is available upon reasonable request from the corresponding author. However, data sets are available as an open source.

ACKNOWLEDGMENTS

The authors would like to acknowledge Mr. Mohammad Kordzanganeh, CEO at Solofied, for clarifying theoretical doubts regarding the QAUM method

REFERENCES

Abbott, B.P., Abbott, R., Abbott, T., Abernathy, M., Acernese, F., Ackley, K. et al. (2016) Observation of gravitational waves from a binary black hole merger. *Physical review letters*, 116(6), 061102.

An, T. (2019) Science opportunities and challenges associated with ska big data. *Science China Physics, Mechanics & Astronomy*, 62, 1–6.

Beniwal, D., Roy, A., Yadav, H. & Chauhan, A. Detection of pulsars by classical machine learning algorithms. In: *2021 2nd International Conference for Emerging Technology (INCET). IEEE, 2021*, pp. 1–7.

de Groot, M. (1977) "Pulsars", by R. N. Manchester and J. H. Taylor. *Irish Astronomical Journal*, 13, 142.

Duchi, J., Hazan, E. & Singer, Y. (2011) Adaptive subgradient methods for online learning and stochastic optimization. *Journal of machine learning research*, 12(7).

Farhi, E., Goldstone, J., Gutmann, S. & Sipser, M. (2000) Quantum computation by adiabatic evolution. *arXiv preprint quant-ph/0001106*,.

Hobbs, G., Lyne, A. & Kramer, M. (2010) An analysis of the timing irregularities for 366 pulsars. *Monthly Notices of the Royal Astronomical Society*, 402(2), 1027–1048.

Hulse, R.A. & Taylor, J.H. (1975) Discovery of a pulsar in a binary system. *Astrophysical Journal, vol. 195, Jan. 15, 1975, pt. 2, p. L51-L53.*, 195, L51–L53.

Janssen, G., Hobbs, G., McLaughlin, M., Bassa, C., Deller, A., Kramer, M. et al. (2014) Gravitational wave astronomy with the ska. *arXiv preprint arXiv:1501.00127*,.

Keane, E., Bhattacharyya, B., Kramer, M., Stappers, B., Bates, S., Burgay, M. et al. (2014) A cosmic census of radio pulsars with the ska. *arXiv preprint arXiv:1501.00056*,.

Kingma, D.P. & Ba, J. (2014) Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*,.

Kordzanganeh, M., Utting, A. & Scaife, A. (2021) Quantum machine learning for radio astronomy. *arXiv preprint arXiv:2112.02655*,.

Lyon, R. (2017) *HTRU2*. UCI Machine Learning Repository, dOI: https://doi.org/10.24432/C5DK6R.

Lyon, R.J. (2016) *Why are pulsars hard to find?* : The University of Manchester (United Kingdom).

Lyon, R.J., Stappers, B., Cooper, S., Brooke, J.M. & Knowles, J.D. (2016) Fifty years of pulsar candidate selection: from simple filters to a new principled real-time classification approach. *Monthly Notices of the Royal Astronomical Society*, 459(1), 1104–1123.

Maoz, D. (2016) *Astrophysics in a Nutshell*. Vol. 16. : Princeton university press.

McLaughlin, M.A. (2013) The north american nanohertz observatory for gravitational waves. *Classical and Quantum Gravity*, 30(22), 224008.

PennyLane (2024) *Quantum gradients with backpropagation*. URL https://pennylane.ai/qml/demos/tutorial_backprop/

PennyLane Documentation (2024) *Gradients and training*. URL https://docs.pennylane.ai/en/stable/introduction/interfaces.html

Powers, D.M. (2020) Evaluation: from precision, recall and f-measure to roc, informedness, markedness and correlation. *arXiv preprint arXiv:2010.16061*,.

Ruder, S. (2016) An overview of gradient descent optimization algorithms. *arXiv preprint arXiv:1609.04747*,.

Schuld, M. & Petruccione, F. (2021) *Machine learning with quantum computers*. : Springer.

Schuld, M., Sweke, R. & Meyer, J.J. (2021) Effect of data encoding on the expressive power of variational quantum-machine-learning models. *Physical Review A*, 103(3), 032430.

Stairs, I.H. (2003) Testing general relativity with pulsar timing. *Living Reviews in Relativity*, 6, 1–49.

Tariq, I., Meng, Q., Yao, S., Liu, W., Zhou, C., Ahmed, A. et al. (2022) Adaboost-dsnn: an adaptive boosting algorithm based on deep self normalized neural network for pulsar identification. *Monthly Notices of the Royal Astronomical Society*, 511(1), 683–690.

Tharwat, A. (2020) Classification assessment methods. *Applied computing and informatics*, 17(1), 168–192.

Verbiest, J., Lentati, L., Hobbs, G., van Haasteren, R., Demorest, P.B., Janssen, G. et al. (2016) The international pulsar timing array: first data release. *Monthly Notices of the Royal Astronomical Society*, 458(2), 1267–1288.

Xiao, J., Li, X., Lin, H. & Qiu, K. (2020) Pulsar candidate selection using pseudo-nearest centroid neighbour classifier. *Monthly Notices of the Royal Astronomical Society*, 492(2), 2119–2127.

Zhou, L., Pan, S., Wang, J. & Vasilakos, A.V. (2017) Machine learning on big data: Opportunities and challenges. *Neurocomputing*, 237, 350–361.