

Ramanujan-Inspired Hybrid Classical–Quantum Classification of Gravitational-Wave Glitches Using the Gravity Spy Dataset

Sakshi Vijay Jadhav* Mr. Palanivel R

School of Information Science, Presidency University, Bengaluru, India

Email:saksuj20@gmail.com

Assistant Professor, Presidency University, Bengaluru, India

Email:jta.palanivel@gmail.com

Abstract—This paper presents a Ramanujan-inspired hybrid classical–quantum framework for gravitational-wave glitch classification using the Gravity Spy dataset. The proposed system combines metadata-based class curation, HDF5 spectrogram loading, statistical image descriptors, and Ramanujan-sum-based row and column signal features. A balanced three-class subset is used for classical classification with support vector machine (SVM), random forest (RF), and XGBoost, while a binary subset is used for a variational quantum classifier (VQC) implemented in PennyLane on a simulator. Experimental results obtained from the notebook show accuracies of 96.00%, 98.00%, and 99.33% for SVM, random forest, and XGBoost respectively, while the binary quantum classifier achieves 75.00% accuracy. The work demonstrates that Ramanujan-inspired compact descriptors are effective for classical and hybrid quantum classification of gravitational-wave glitches.

Index Terms—Gravity Spy, glitch classification, Ramanujan-inspired feature extraction, hybrid classical–quantum learning, variational quantum classifier, gravitational-wave detector characterization.

I. INTRODUCTION

Gravitational-wave detectors are extremely sensitive scientific instruments and therefore capture a variety of non-astronomical transient artifacts in addition to useful scientific information. These transient disturbances, commonly called glitches, degrade detector quality and may interfere with downstream analysis pipelines. Automatic classification of glitches is therefore important for detector characterization, quality control, and trustworthy gravitational-wave data analysis. Recent 2026 work in hybrid quantum-classical learning, dimensionality reduction, robust quantum classifiers, and practical benchmarking provides a useful computational context for studying compact feature-driven classification pipelines [1]–[4].

The Gravity Spy project offers a labeled glitch repository that converts detector disturbances into spectrogram-like visual patterns, making the problem suitable for pattern recognition and machine learning. In this paper, a hybrid classical–quantum workflow is proposed in which handcrafted Ramanujan-inspired features are extracted from glitch spectrograms and then used by classical and quantum classifiers. The intent is to build a project that is experimentally meaningful

and academically presentable, with diagrams, feature visualizations, confusion matrices, and performance comparison tables directly tied to the implementation notebook.

II. REVIEW OF LITERATURE

Recent 2026 studies show that quantum and hybrid learning pipelines are increasingly being evaluated on practical classification, compression, and robustness tasks. Franco *et al.* used SHAP-guided feature selection for quantum phase classification and showed that carefully reduced feature sets can improve QML behavior on compact inputs [2]. Adermann *et al.* studied variational quantum learning under quantum error detection and highlighted the importance of reliable circuit design in NISQ-era workflows [3]. Dowling *et al.* further examined the robustness of quantum classifiers, strengthening the case for careful evaluation beyond raw accuracy [4].

Several 2026 papers are especially relevant to the present project setup. Majumdar proposed quantum-stable robust principal component analysis for noisy regimes [5], while Jabeen *et al.* explored quantum machine learning from classical data representations [6]. Cirillo *et al.* benchmarked QML methods under realistic noise conditions [7], and Alharbi reported a modified transfer-learning-based variational quantum circuit for image classification [8]. Tripathi *et al.* demonstrated a hybrid quantum-classical predictive study on collision-event analysis [1], which is structurally similar to the present work in the sense that classical data preparation and reduced feature representations are paired with quantum learning models. These studies motivate the use of compact Ramanujan-inspired descriptors, PCA-reduced quantum inputs, and comparative classical baselines in the present Gravity Spy classification framework.

III. METHODOLOGY

3.1 System Architecture

The overall system architecture is shown in Fig. 1. The pipeline begins with two data inputs: the Gravity Spy metadata and the HDF5 spectrogram image repository. The data are then passed through class selection and balancing, after which a feature engineering module computes basic image statistics

along with Ramanujan-inspired descriptors derived from row-wise and column-wise signals. The resulting feature vectors feed both the classical branch and the quantum branch. The classical branch trains SVM, random forest, and XGBoost, while the quantum branch applies PCA and then a variational quantum classifier. Both branches are finally compared using classification metrics.

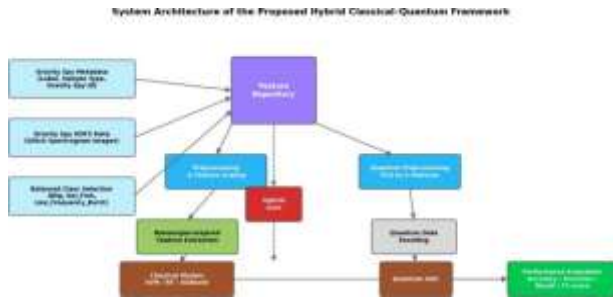


Fig. 1. System architecture of the proposed hybrid classical–quantum Gravity Spy glitch classification framework.

3.2 Model Setup

For the classical setting, the selected glitch classes are `Blip`, `Koi_Fish`, and `Low_Frequency_Burst`. A balanced subset is formed with 250 samples per class, resulting in approximately 750 total instances. For the quantum setting, the binary subset uses `Blip` and `Koi_Fish`, with 120 samples per class to keep simulation feasible.

The classical models are configured as follows:

- **SVM:** RBF kernel, $C = 10$, $\gamma = \text{scale}$, probability enabled.
- **Random Forest:** 300 estimators.
- **XGBoost:** 300 estimators, max depth 6, learning rate 0.1, subsample 0.9, column sample by tree 0.9.

The quantum model uses:

- 4 PCA features,
- 4 qubits,
- 2 variational layers,
- Adam optimizer with learning rate 0.05,
- 15 training epochs on a simulator.

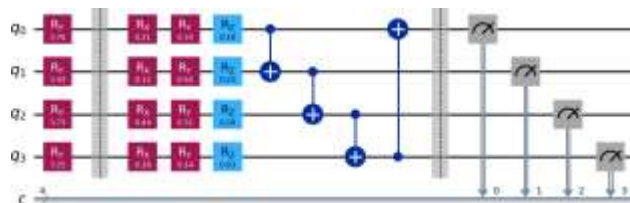


Fig. 2. Quantum circuit diagram used for presentation and explanation of the variational quantum classifier.

IV. ALGORITHM

The algorithmic workflow of the proposed method is summarized below. The sequence is directly based on the Google

Colab implementation, starting from dataset acquisition and ending with classical–quantum comparative evaluation. In addition to loading and balancing the selected Gravity Spy classes, the workflow explicitly constructs Ramanujan-inspired descriptors, performs scaling for the classical branch, and applies PCA reduction for the quantum branch. This makes the algorithm section consistent with the actual notebook cells rather than a generic summary, and it clarifies how the same extracted features support both the multi-class classical pipeline and the smaller simulator-based binary quantum experiment.

From an implementation perspective, the algorithm is designed to preserve a direct correspondence between each notebook stage and each analytical component of the paper.

The metadata loading, balancing, HDF5 image access, hand-crafted feature generation, label encoding, classical training, and quantum simulation steps are therefore not independent modules; rather, they operate as one continuous pipeline in which each stage prepares the next stage for evaluation. This end-to-end sequence is important because it demonstrates that the reported results are produced from a reproducible computational workflow rather than from isolated experiments.

The same algorithmic chain also explains how the proposed framework remains internally consistent across the classical and quantum branches. Once the curated image subset is converted into compact Ramanujan-inspired descriptors, the downstream learning stages no longer depend on raw spectrogram storage or ad hoc manual intervention. This continuity is important for the present project because it ensures that feature generation, classifier training, and final result reporting all emerge from a single reproducible notebook workflow.

In practical terms, the algorithm also reflects the computational trade-off between interpretability and complexity. Instead of relying on a very deep end-to-end neural model, the workflow uses carefully designed feature extraction, explicit normalization, and a structured split between classical and quantum training. This makes the sequence easier to validate academically, easier to reproduce during demonstration, and more suitable for a student project where clarity of implementation is as important as final predictive performance.

Algorithm 1: Ramanujan-Inspired Hybrid Gravity Spy Classification

- 1) Load Gravity Spy metadata and select target glitch classes.
- 2) Build a balanced subset for the chosen classes.
- 3) Open the HDF5 dataset and load the selected glitch images.
- 4) For each image, compute basic image descriptors.
- 5) Form row-wise and column-wise summary signals.
- 6) Compute the Ramanujan sum

$$c_q(n) = \sum_{\substack{1 \leq a \leq q \\ \gcd(a, q)=1}} \cos \frac{2\pi an}{q}$$

for the selected values $q \in \{2, 3, 4, 5, 6\}$.

- 7) Compute Ramanujan-inspired descriptors from these signals.

- 8) For each signal $x(n)$, evaluate

$$R_q = \frac{\left| \sum_n x(n) c_q(n) \right|}{|x| |c_q| + \epsilon}$$

and append the resulting descriptors to the feature vector.

- 9) Concatenate all descriptors into a final feature vector.
- 10) Encode the labels numerically.
- 11) Split the classical feature set into train and test partitions.
- 12) Normalize the classical features and train SVM, RF, and XGBoost.
- 13) Build a binary quantum subset.
- 14) Normalize and reduce the quantum features using PCA.
- 15) Train the variational quantum classifier.
- 16) Evaluate all models using accuracy, precision, recall, F1-score, and confusion matrices.

- scikit-learn for train-test split, scaling, SVM, random forest, PCA, and evaluation,
- xgboost for boosted tree classification,
- PennyLane for the variational quantum classifier,
- Qiskit for quantum circuit visualization,
- matplotlib and seaborn for figures and heatmaps.

The direct data sources used by the notebook are:

- trainingset_v1d0_metadata.csv,
- trainingsetv1d0.h5.

Within the Colab workflow, the metadata are first inspected to identify the available glitch classes and their frequency. A curated balanced subset is then created, after which the HDF5 file is opened and the selected spectrogram images are read. The notebook defines the Ramanujan-inspired feature extraction functions, constructs the feature matrix, applies scaling and label encoding, and then trains the classical baselines. For the quantum branch, a smaller binary subset is created and reduced to four principal components before variational circuit training.

VI. RESULTS

6.1 Model Performance

The notebook produced the following measured accuracies on the curated dataset:

- SVM: 96.00%
- Random Forest: 98.00%
- XGBoost: 99.33%
- Quantum VQC: 75.00%

Since the classical branch is multi-class and the quantum branch is binary, the comparison is intended as a practical hybrid study rather than a perfectly identical benchmark. The detailed metric summary is shown in Table I.

The measured results indicate that the handcrafted feature space is highly informative for classical classification. XGBoost achieves the strongest performance, followed closely by random forest and then SVM. This suggests that the proposed Ramanujan-inspired descriptors capture sufficient discriminative structure from the glitch spectrograms for conventional machine learning models to separate the selected classes reliably.

TABLE I
MODEL PERFORMANCE SUMMARY

Model	Accuracy	Precision	Recall	F1-score
SVM	0.9600	0.9600	0.9600	0.9600
Random Forest	0.9800	0.9800	0.9800	0.9800
XGBoost	0.9933	0.9933	0.9933	0.9933
Quantum VQC	0.7500	0.8000	0.7500	0.7400

6.2 Confusion Matrix (Table and Figure)

The class-wise behavior of the models is illustrated through confusion matrices. For the classical setting, three confusion matrices were generated for SVM, random forest, and XGBoost. The quantum branch produced a separate binary confusion matrix. The summarized interpretation of the quantum

Workflow Diagram of Gravity Spy Glitch Classification



Fig. 3. Workflow diagram of the proposed Gravity Spy glitch classification pipeline.

V. ENVIRONMENT SETUP

The implementation was developed in a Python notebook environment. The core software libraries are:

- numpy, pandas, and h5py for numerical computation and HDF5 handling,

confusion matrix is reported in Table II, while the combined classical confusion-matrix visualization is shown in Fig. 4.

These matrices provide a more detailed view of decision behavior. The SVM model shows a small amount of confusion involving Low_Frequency_Burst, whereas random forest reduces the error pattern and XGBoost provides the cleanest separation among the three classes. The binary quantum model shows asymmetric behavior, recognizing Koi_Fish more strongly than Blip, which is consistent with its reduced feature space and limited variational depth.

TABLE II
SUMMARY OF QUANTUM CONFUSION MATRIX INTERPRETATION

Item	Observation
True negatives	13 Blip samples were correctly predicted as Blip.
False positives	11 Blip samples were incorrectly predicted as Koi_Fish.
False negatives	1 Koi_Fish sample was incorrectly predicted as Blip.
True positives	23 Koi_Fish samples were correctly predicted as Koi_Fish.
Interpretation	The quantum classifier preserves strong recall for Koi_Fish after PCA compression, but the lower recall for Blip indicates that some discriminative information is lost in the reduced four-feature representation.

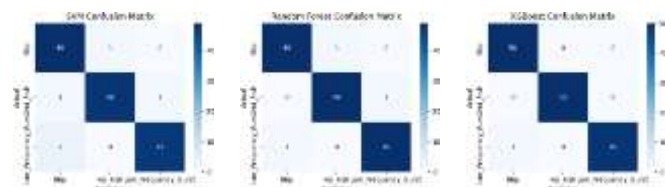


Fig. 4. Confusion matrices of SVM, random forest, and XGBoost on the curated Gravity Spy multi-class subset.

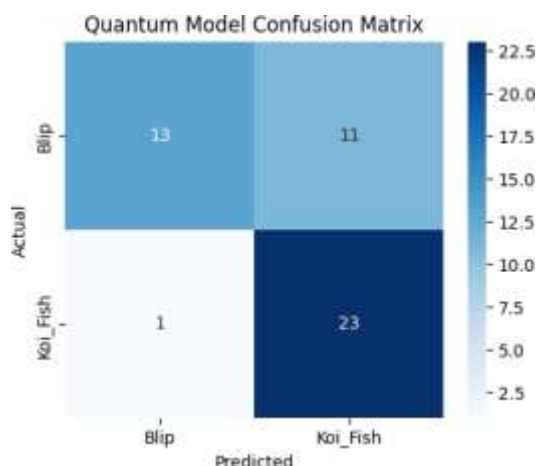


Fig. 5. Confusion matrix of the variational quantum classifier on the binary Gravity Spy subset.

6.3 Prediction Based on Glitches Gravity Spy Classification

The proposed model predicts the glitch category from a spectrogram image after extracting Ramanujan-inspired and statistical features. Example glitch images used for qualitative validation are shown in Figs. 6 and 7. These figures provide a direct visual connection between the selected Gravity Spy classes and the classification outputs discussed in the notebook results.

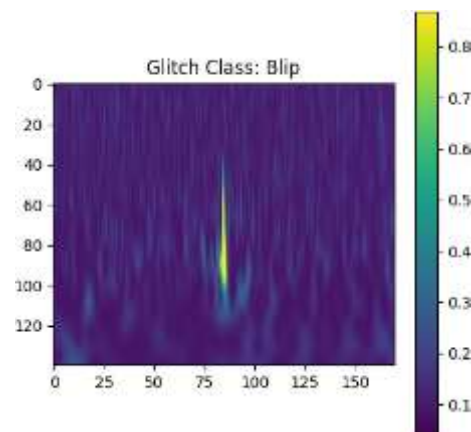


Fig. 6. Single example glitch image used for inspection and feature extraction.

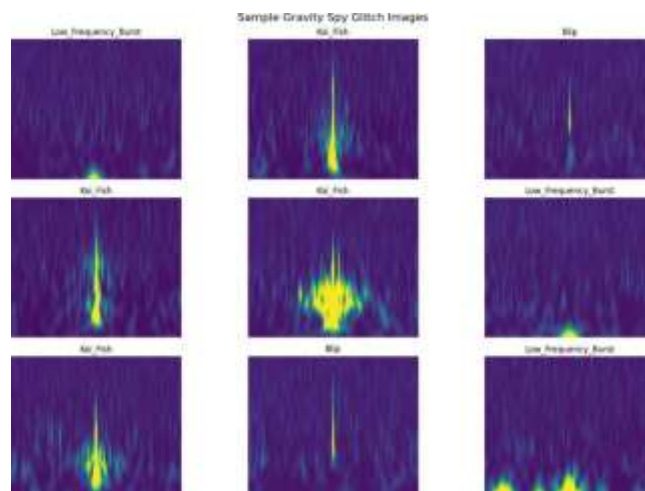


Fig. 7. Grid of representative Gravity Spy glitch samples from the selected classes.

For unseen predictions, the same feature-extraction procedure can be applied to a new Gravity Spy image. The trained classical model then returns one of the learned glitch labels, while the quantum branch predicts between the two binary quantum classes.

From a deployment perspective, this means that a new glitch spectrogram follows the same sequence: HDF5 loading, feature extraction, scaling, and model inference. Thus, the project is not restricted to retrospective benchmarking; it can also be used as a practical classification pipeline for newly observed glitch samples from the selected categories.

6.4 Comparative Analysis with Baseline Algorithm

The baseline comparison clearly favors the classical models in this project setting. XGBoost achieved the highest accuracy at 99.33%, followed by random forest at 98.00% and SVM at 96.00%. The variational quantum classifier reached 75.00% on the binary subset. These findings suggest that handcrafted Ramanujan-inspired descriptors are highly effective for classical glitch classification and still usable for a lightweight quantum proof of concept.

This comparison also shows that the same feature engineering strategy can support both conventional and quantum learning pipelines. While the quantum branch does not yet match the strongest classical baseline, it demonstrates that reduced Ramanujan-inspired representations can be integrated with PCA and variational circuits to produce meaningful predictions in a simulator setting. The classical baselines remain stronger primarily because they operate on a richer multi-class task with models better suited to the present feature dimensionality.

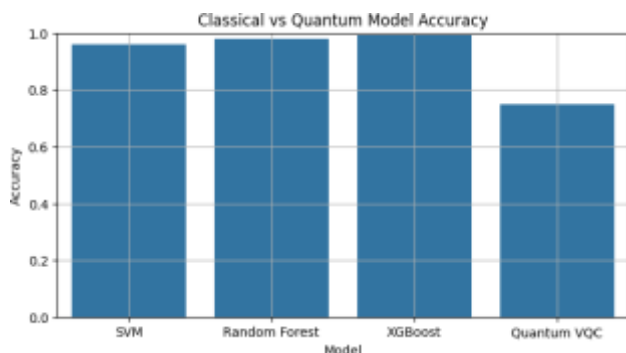


Fig. 8. Accuracy comparison of the classical and quantum models.

6.5 Limitations

The present framework should be interpreted as a controlled project-level study rather than a full production benchmark for detector characterization. The classical and quantum branches do not solve exactly the same task, since the classical models are trained on a three-class subset while the quantum model is restricted to a binary subset to keep simulator training feasible. The proposed Ramanujan-inspired descriptors are intentionally compact and interpretable, but this handcrafted representation may omit fine-grained spectrogram structures that richer learned feature extractors could preserve. In addition, the quantum pipeline depends on PCA compression to four features before variational encoding, and that aggressive reduction likely contributes to the weaker Blip recall observed in the binary confusion matrix. Finally, all quantum experiments are performed on a simulator, so the reported quantum results demonstrate feasibility of the hybrid workflow rather than a verified hardware-level advantage on the full Gravity Spy archive.

VII. CONCLUSION

This paper presented a Ramanujan-inspired hybrid classical–quantum framework for glitch classification using the Gravity Spy dataset. The final implementation followed a complete project structure that includes data loading, review of literature, methodology, system architecture, model setup, algorithm design, environment setup, result analysis, confusion-matrix interpretation, prediction logic, and comparative baseline analysis. The results show that the classical branch performs very strongly on the curated multi-class subset, with XGBoost achieving the highest accuracy, while the quantum branch demonstrates a feasible hybrid approach for binary glitch classification.

The inclusion of system diagrams, workflow diagrams, glitch sample images, feature vector plots, PCA analysis, circuit diagrams, comparison tables, and confusion matrices makes the framework suitable for a full project report and IEEE-style paper submission. Future work may extend the binary quantum setup to multiclass quantum classification, larger glitch subsets, and more advanced explainability strategies.

REFERENCES

- [1] S. Tripathi, H. Upadhyay, and J. Soni, “A quantum machine learning-based predictive analysis of CERN collision events,” *Scientific Reports*, vol. 16, p. 682, 2026.
- [2] G. S. Franco, F. Mahlow, P. M. Prado, G. E. L. Peixe, L. A. M. Rattighieri, and F. F. Fanchini, “Quantum phases classification using quantum machine learning with SHAP-driven feature selection,” *Quantum Information Processing*, vol. 25, p. 106, 2026.
- [3] E. Adernann, H. Suzuki, and M. Usman, “Variational quantum machine learning with quantum error detection,” *Quantum Machine Intelligence*, vol. 8, p. 1, 2026.
- [4] N. Dowling, M. T. West, A. Southwell *et al.*, “Adversarial robustness guarantees for quantum classifiers,” *npj Quantum Information*, vol. 12, p. 16, 2026.
- [5] A. Majumdar, “Quantum-stable robust principal component analysis: theory and evidence from NISQ regimes,” *Quantum Machine Intelligence*, vol. 8, p. 6, 2026.
- [6] S. Jabeen, D. Kurdydyk, A. Palnitkar *et al.*, “Quantum machine learning for state tomography using classical data,” *Quantum Machine Intelligence*, vol. 8, p. 40, 2026.
- [7] F. Cirillo, C. Esposito, and J. T. Seo, “Benchmarking quantum machine learning methods for intrusion detection on noisy quantum computers,” *Quantum Machine Intelligence*, vol. 8, p. 27, 2026.
- [8] M. Alharbi, “Modified transfer learning based variational quantum circuit for MNIST image classification,” *Connection Science*, vol. 38, no. 1, 2026.
- [9] A. Engelsberger, M. Pšenic’ková, and T. Villmann, “Prototype-based classifiers and vector quantization on a quantum computer—implementing integer arithmetic oracles for nearest prototype search,” *Entropy*, vol. 28, no. 2, p. 229, 2026.
- [10] G. Sergioli, C. Cuccu, C. S. Rieger, A. C. G. Arango, B. K. Behera, R. Era, and R. Giuntini, “A quantum-inspired classification for random mixed states,” *Scientific Reports*, vol. 16, p. 10668, 2026.
- [11] A. Khrennikov, “Quantum-like cognition and AI toward the convergence of natural and artificial intelligence,” *Discover Artificial Intelligence*, vol. 6, p. 167, 2026.
- [12] P. Jain, P. Pathak, K. Bhatia *et al.*, “Quantum-classical generative models for drug design,” *Quantum Machine Intelligence*, vol. 8, p. 30, 2026.
- [13] T. Schwägerl, Y. Chai, T. Hartung *et al.*, “Benchmarking variational quantum algorithms for combinatorial optimization in practice,” *Quantum Machine Intelligence*, vol. 8, p. 26, 2026.
- [14] R. Nembrini, M. F. Dacrema, and P. Cremonesi, “Minor embedding for quantum annealing with reinforcement learning,” *Quantum Machine Intelligence*, vol. 8, p. 15, 2026.

- [15] S. Tarquini, D. Dragoni, M. Vandelli *et al.*, “Testing quantum and simulated annealers on the drone delivery packing problem,” *Quantum Machine Intelligence*, vol. 8, p. 14, 2026.
- [16] R. Palanivel and P. Muthulakshmi, “Error mitigation using quantum neural Q network in secure qutrit distribution on Cleve’s protocol on quantum computing,” *Quantum Information Processing*, vol. 23, p. 147, 2024.
- [17] R. Palanivel and P. Muthulakshmi, “Design and analysis of quantum transfer fractal priority replay and mirdad priority loss algorithms for quantum reinforcement learning,” in *Data Management, Analytics and Innovation*, ser. Lecture Notes in Networks and Systems. Singapore: Springer, 2024, vol. 997.
- [18] R. Palanivel and P. Muthulakshmi, “Design and analysis of parallel quantum transfer fractal priority replay with dynamic memory algorithm in quantum reinforcement learning for robotics,” *IET Quantum Communication*, 2024.
- [19] R. Palanivel and P. Muthulakshmi, “Quantum prioritized experience replay with MaDi-based priority and quantum circuit mechanisms for optimizing reinforcement learning,” *International Journal of Advanced Technology and Engineering Exploration*, vol. 11, no. 121, pp. 1664–1680, 2024.