Time Series Anomaly Detection Using Hybrid Quantum Variational Rewinding

Priyavrata Tiwari

***Abstract*—Time series data is a fundamental characteristic of many systems with both academic and industrial significance. Deviations from expected temporal behavior in such data can reveal anything from system malfunctions to previously undocumented features. Consequently, developing efficient techniques and optimizing existing methods for time series anomaly detection (TAD) is an important area of research. A quantum approach for TAD, termed Quantum Variational Rewinding (QVR), is described in [1]. The algorithm's reliance on heuristic parameter selection and the need to choose among various normal probability distributions have been addressed by integrating a conventional feed-forward neural network, providing a more systematic approach. Additionally, a conventional Quantum Variational circuit, which is simpler to implement than the original QVR circuit, has been tested to determine whether the algorithm's performance from [1] can be replicated with minor adjustments. This study presents a comparative analysis of different Quantum-Classical hybrid architectures, each leveraging QVR as a foundation, to assess and benchmark performance across varied configurations.**

I Introduction

Anomaly detection is of great importance across diverse fields, from industry to academia. In traditional machine learning approaches, models are trained on normal (non-anomalous) time series data, with anomalies subsequently identified by their deviations from the learned normal patterns. This approach is widely applicable, with prominent use cases in financial fraud detection, medical diagnostics, and astronomy, among others.

Popular architectures for time-series anomaly detection (TAD) include Recurrent Neural Networks (RNNs), Long Short-Term Memory (LSTM) networks, and Bidirectional Long Short-Term Memory (Bi-LSTM) networks. These architectures are often further combined with other network topologies, resulting in more complex, high-performance models that excel in handling large datasets. Such hybrid networks have proven particularly effective in improving anomaly detection accuracy and robustness across various domains.
The Quantum Variational Recurrent (QVR) method employs time devolution instead of traditional evolution to perform time-series anomaly detection (TAD), with a detailed summary provided later in the text. In this algorithm, parameters are selected heuristically, which, while potentially useful in some cases, may lack reliability. The algorithm attempts to learn a set of means and standard deviations that represent normal distributions from which the parameters for the unitary operators are sampled and subsequently utilized in the circuit. This introduces a probabilistic nature to the algorithm.

The aforementioned limitations can be addressed by incorporating Conventional Feedforward Neural Networks (FFNs). This integration reduces the number of direct parameters that the variational circuits must learn, thereby enhancing their robustness to noise by minimizing the degrees of freedom that could be negatively impacted by noise.

In this approach, the FFN is positioned prior to the QVR architecture, allowing for the pre-processed data, rather than raw data, to be fed into the architecture. Training then continues with this refined dataset, thereby potentially improving the performance and reliability of the overall model.

A. *Quantum Variational Rewinding (QVR)*

The Quantum Variational Recurrent (QVR) approach offers a quantum-based alternative to conventional machine learning methods within the framework of Quantum Machine Learning (QML). This algorithm performs time-series anomaly detection (TAD) by harnessing the power of variational quantum circuits and unitary time-evolution operators. Specifically, parametrized unitary operators are trained to evolve quantum states that encode time-series data representing the normal behavior of a system.

During training, the parameters of the unitary operators are optimized so that the resulting quantum states yield expectation values of an observable that cluster around a central point. When a new time series is input, the model generates expectation values that either lie within a threshold distance from this learned center (indicating normal behavior) or exceed this distance (signaling an anomaly). Thus, the QVR algorithm effectively distinguishes between normal and anomalous time series based on quantum state deviations.

The algorithm starts with the training set X having N number of d-dimensional real world time series, $x_i(t) = \{x_i^1(t), x_i^2(t), x_i^3(t)...x_i^d(t)\}$ where $x_i(t)$ is the i^{th} time series in X. During each training iteration (mini-batch iteration), **m** number of time series are sampled randomly (in [1], m = 50) with τ = $\{t_1, t_2, t_3...t_p\}$ randomly chosen time points for each of the time series. With these mini-batches we prepare the quantum states using angle encoding as $U[x_i(t_j)]\,|0\rangle^{\bigotimes n}$ = $|x_i(t_j)\rangle$. The number of qubits n = dimensions of the time series in the data = d. And unitary matrix U (in the work) = $\bigotimes_{k=1}^{d} R_y(x_i^k(t_j))$ where $R_y(\theta)$ is rotation matrix, representing rotation of the qubit around y-axis by an angle θ. Each of the quantum states thus formed are then undergone a dynamic process represented by a general parametrized Hamiltonian $H(\alpha, \epsilon)$ given as

$$|x_i(t_j), \alpha, \epsilon\rangle = e^{-iH(\alpha,\epsilon)t_j}\,|x_i(t_j)\rangle \tag{1}$$

using the eigenvalue decomposition we have

$$e^{-iH(\alpha,\epsilon)t_j} = W^{\dagger}(\alpha)D(\epsilon, t_j)W(\alpha) \tag{2}$$

Where α and ϵ are the parameters learned by the algorithm. This is conventionally seen as a forward time evolution operator, but in the context of the alogrithm this has been seen as devolving $|x_i(t_j)\rangle$ by time t_j. This is allowed as a forward time evolution given by $H(\alpha,\epsilon)$ is equivalent to a devolution represented by $-H(\alpha,\epsilon)$ as this is also a valid Hamiltonian. So a quantum state $|x_i(t_j)\rangle$ is created and then is time devolved to by t_j. The ϵ is uniformly randomly drawn from independent normal distribution $e_q \sim N(\mu_q,\sigma_q)$, Where q ranges from 1 to 2^n-1, thus forming a vector $\epsilon \sim N(\mu,\sigma)$ as parameter whics is learned by the model. After a single random ϵ we evaluate the expectation value of a general parametrized observable O_n as

$$\Omega(x_i(t_j),\alpha,\epsilon,\eta) = \langle x_i(t_j),\alpha,\epsilon|\, O_n\, |x_i(t_j),\alpha,\epsilon\rangle \tag{3}$$

Where η is a parameter vector $(eta_0, \eta_1, \eta_2 ... \eta_g)$ and O_n is a n qubit operator in the work [1] $O_n = \eta_0 I - \frac{1}{n}\sum_{i=1}^{n} \eta_i \sigma_z^i$. Next the classical expectation values of the the square of Eq. 3 drawging $\epsilon \sim N(\mu,\sigma)$ and rescaled by L as

$$C_1(x_i(t_j),\theta) = \frac{\mathbb{E}[\Omega^2(x_i(t_j,\alpha,\epsilon,\eta))]}{L} \tag{4}$$

Where $\theta = [\alpha,\mu,\sigma,\eta]$ Where L has been chosen heuristically to be 4. The total contribution of one time series in the loss function is given as

$$C_2(x_i,\theta) = \frac{\sum_{t_j \epsilon B_\tau} C_i(x_i(t_j),\theta)}{N_\tau} \tag{5}$$

Where N_{tau} is the number of time points chosen for each of the time series in the training process (50 in our case). The task is to find parameters that cluster all instances of the Eq. 5 about a given point which with our choice of O_n is clustering about a center η_0. The total loss function for one mini-batch iteration is given as

$$C(\theta) = P_\tau(\sigma) + \frac{\sum_{x_i \epsilon B_x} C_2(x_i(t),\theta}{2N_x} \tag{6}$$

where N_x is the total time series taken in one mini-batch(50 in our case) and $P_\tau(\sigma)$ is a regularization function designed to penalize large entries of σ, and is given as

$$P_\tau(\sigma) = \frac{\sum_{m=1}^{Q} tan^{-1}(2\pi\tau_m|\sigma_m|)}{\pi Q} \tag{7}$$

Where $\tau_1 = \tau_2 = ... = \tau_{Q-1} = \tau$ for a single contraction hyper parameter τ. This ensures $C(\theta)$ takes values between 0 and 1. This loss function is further minimized by the model during training to obtain

$$\theta^* = argmin_\theta[C(\theta] \tag{8}$$

After a good approximation of θ^* the model calculates the anomaly score of a unseen time series as

$$a_X[y] = |2C(\theta^*) - 2P_\tau(\sigma) - C_2[y,\theta^*]| \tag{9}$$

if the anomaly score ($a_X[y]$) exceeds a threshold ζ the time series is classified as anomalous other wise as normal. ζ is chosen to maximize any performance matrix during the validation phase, using a validation set of labeled anomalous time series.

II Neural Network at the beginning of QVR

Adding a neural network at the beginning of the QVR model, we can potentially replace the normal distributions with outputs from the neural network. The network would serve to generate parameter values dynamically, eliminating the need for independent sampling from normal distributions. This may allow the model to adapt these initial values based on learned features from the data rather than static, stochastic initialization, which could lead to better overall optimization and anomaly detection accuracy. The neural network consists of an input layer with 50 neurons (corresponding to the 50 time points of a time series being fed into the architecture), followed by three hidden layers with 128, 64, and 32 neurons, respectively. The output layer contains 7 neurons. The output from these 7 neurons is then fed into the modified QVR circuit. In the analyzed architecture, the part involving the normal distribution has been removed. Instead, the preprocessed data is now fed into the Quantum Variational Representation (QVR) architecture. The underlying probability distribution, which was initially learned by the QVR, is now learned by the neural network. The loss function has been modified accordingly, and is given by:

$$C_1(x_i,\theta) = \frac{1}{N_t}\sum_{j=1}^{N_t} \langle x_i(t_j)|\, O_n\, |x_i(t_j)\rangle \tag{10}$$

Here,θ represent the parameters of the variational circuit, C_1 represents the contribution to the loss function from one time series, and N_t is the number of time points in a single time series. The loss function is given by:

$$L = \frac{1}{N_x}\sum_{i=1}^{N_x} C_{1i} \tag{11}$$

Where N_x number of time series in one mini batch. As the normal distribution part of the algorithm has been removed the regularization function. The anomaly score for the architecture changes to

$$a_x[y] = |L(\theta^*) - C_1[y,\theta^*]| \tag{12}$$

Where θ^* is the optimized parameters. Figure[1] givs a schematic of the architecture.

III Using Only Variational Circuit

The analysis of the QVR algorithm does not directly leverage temporal properties in terms of sequential or recurrent processing across multiple time steps. Since the algorithm does not predict the time series for future points and only clusters a time series around a given point, a potential approach would be to replace the time-dependent unitary gates with static unitary gates along with a layer of entangling gates that have trainable parameters, while retaining the same loss function given in equation [Fig 6]. A schematic of the architecture is given in [Fig 2]

Following can be the potential impact on the model in this case

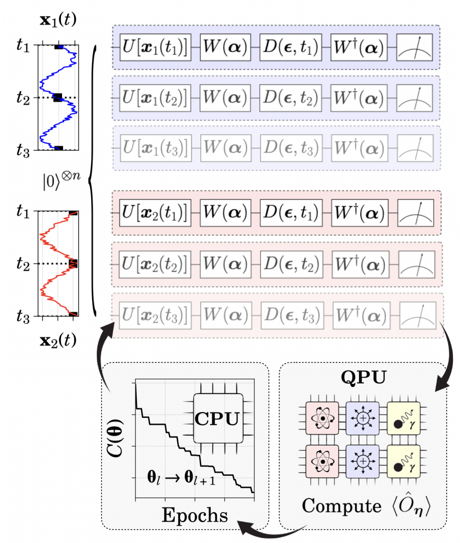
$\mathbf{x}_1(t)$
t_1
t_2
t_3
$|0\rangle^{\otimes n}$
$\mathbf{x}_2(t)$
$U[\boldsymbol{x}_1(t_1)]$ $W(\boldsymbol{\alpha})$ $D(\boldsymbol{\epsilon}, t_1)$ $W^\dagger(\boldsymbol{\alpha})$
$U[\boldsymbol{x}_1(t_2)]$ $W(\boldsymbol{\alpha})$ $D(\boldsymbol{\epsilon}, t_2)$ $W^\dagger(\boldsymbol{\alpha})$
$U[\boldsymbol{x}_1(t_3)]$ $W(\boldsymbol{\alpha})$ $D(\boldsymbol{\epsilon}, t_3)$ $W^\dagger(\boldsymbol{\alpha})$
$U[\boldsymbol{x}_2(t_1)]$ $W(\boldsymbol{\alpha})$ $D(\boldsymbol{\epsilon}, t_1)$ $W^\dagger(\boldsymbol{\alpha})$
$U[\boldsymbol{x}_2(t_2)]$ $W(\boldsymbol{\alpha})$ $D(\boldsymbol{\epsilon}, t_2)$ $W^\dagger(\boldsymbol{\alpha})$
$U[\boldsymbol{x}_2(t_3)]$ $W(\boldsymbol{\alpha})$ $D(\boldsymbol{\epsilon}, t_3)$ $W^\dagger(\boldsymbol{\alpha})$
$C(\boldsymbol{\theta})$
CPU
$\boldsymbol{\theta}_l \rightarrow \boldsymbol{\theta}_{l+1}$
Epochs
QPU
Compute $\langle \hat{O}_\eta \rangle$

Fig. 1. Schematic of the QVR-time series anomaly detection[1]

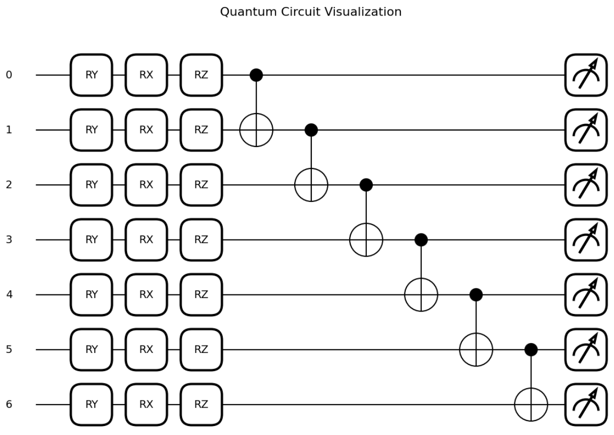

Fig. 2. Schematic of the variational circuit with six qubits

- **Reduced Complexity and Circuit Depth:** Removing the time-dependent unitary reduces the number of parameters and the need for potentially complex implementations of $D(\epsilon, t_j)$ across multiple time points. This has lead to a reduction in circuit depth, which is particularly advantageous for current noisy intermediate-scale quantum(NISQ) devices where deep circuits can introduce significant noise and decoherence.
- **Inter feature relationships:** Introduction of the entangling layers ensure that if any inter feature relations are present in the time series, they can be reflected into the model,capturing the multivariate dependencies within a time series.
- The model is more efficient to implement and trains faster compared to the original QVR algorithm implementation.

The weights of the variational circuit have been initialized using a uniform normal distribution. While the importance of weight initialization for model convergence has been discussed in previous studies, this aspect has not been explored in the present work. It could, however, be considered as part of future research.

Another problem which was encountered during the training of this model was of barren plateau where the gradients of the loss function with respect to the parameters of a quantum circuit become very small, effectively causing the optimization process to stall or slow down significantly [3].

This issue has been mitigated using the method of **partial measurement**[2], where instead of measuring all the qubits in each iteration, we randomly measure a subset of qubits and use this to construct our loss function.

IV Training and Validation of the models

The models have been trained over the time series of Cosmic Microwave background with Seven features have been chosen. The features were collected independently and are arranged in a time series. Following are the features of the dataset

- Temperatures
- Luminosities
- Spectral Densities
- Optical Depth
- Baryon Densities
- Hubble Parameters
- Dark Matter Density

This highlights the need for using a 7-qubit variational circuit. Identifying anomalies in this dataset is crucial, as such regions can be studied in greater detail to understand the underlying causes of the anomalies. The following points outline the training process:

- The time series containing 10,000 time points of non-anomalous data is divided into 200 smaller time series, each consisting of 50 time points.
- In each iteration, 50 time series are randomly selected and fed into the network.
- The loss function is minimized after each mini-batch iteration. The neural network model and the QVR were trained for 400 epochs, while the variational circuit model achieved the same loss value after just 100 epochs. Beyond 100 epochs, the variational circuit model began to overfit. This conclusion is being made as the validation accuracy decreases with increase in the number of epochs, instead of continuing to learn the general trend[Fig 3].

A. *Validation*

In the validation process, 50 non-anomalous time series and 50 labeled anomalous time series are fed into the architecture, and anomalous scores are obtained. Based on these scores, predictions are made for all the validation time series (to determine whether they are anomalous or not). The parameter ζ is then determined through grid search to optimize the balanced accuracy score.

Balanced Accuracy Score = $\frac{TP+TN}{TN+FP}$

Where:

- TP: True Positives

- TN: True Negatives
- FN: False Negatives
- FP: False Positives

The balanced accuracy score is used as non-anomalous time series significantly outnumber the anomalous ones. The metric is designed to give equal weight to the performance on each class, regardless of their frequency. When an unseen time series is fed into the circuit, the anomaly score is calculated. If the score exceeds the threshold value of ζ the time series is marked as anomalous

V Results

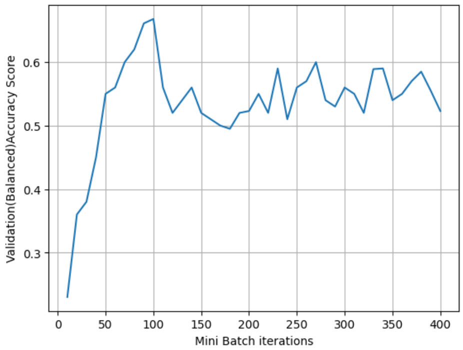

Fig. 3. Variation of Validation accuracy with number of mini-batch iterations for Variational circuit model

A. Training Losses

The figures show the variation of loss function associated with all the architectures

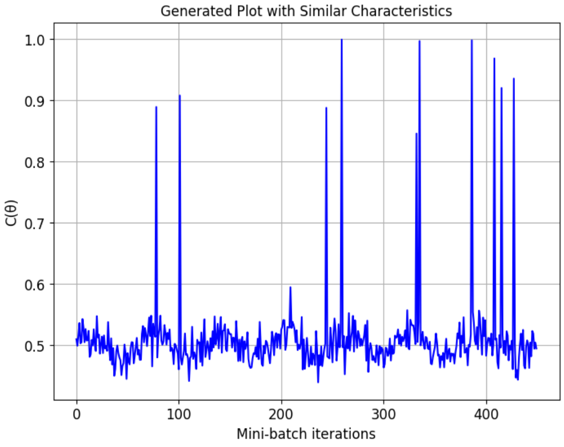

Fig. 4. Loss Function in QVR

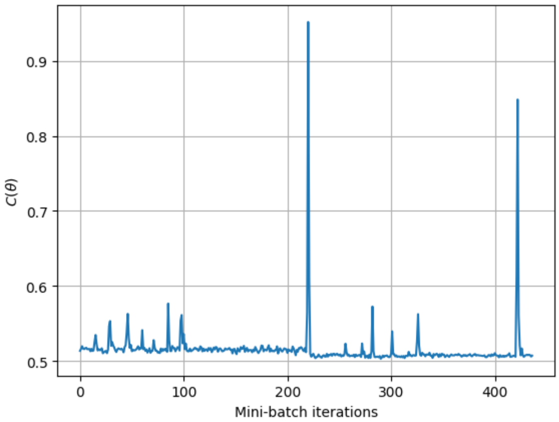

Fig. 5. Loss Function in Neural Layer Circuit

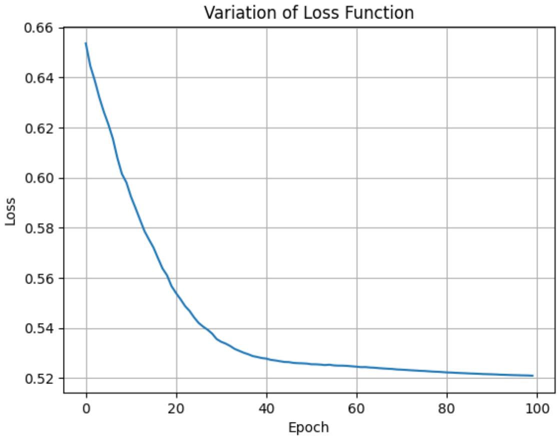

Fig. 6. Loss Function in Variational Circuit(Epoch=Mini-Batch iterations)

B. Validation Results

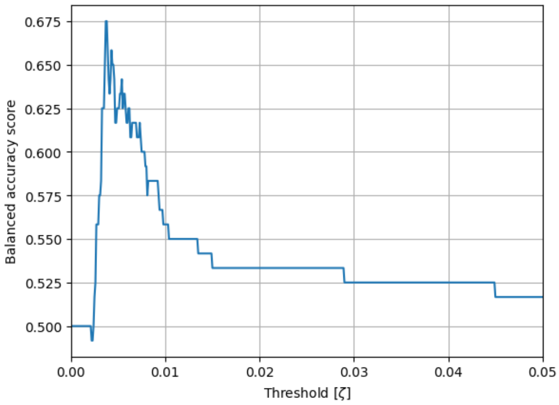

Fig. 7. Validation performance for QVR

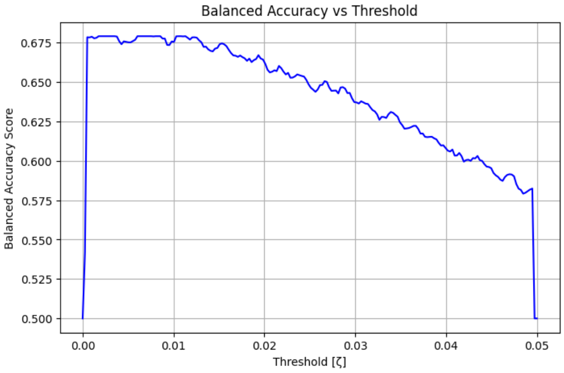

Fig. 8. Balanced Accuracy Score for Neural Network Architecture

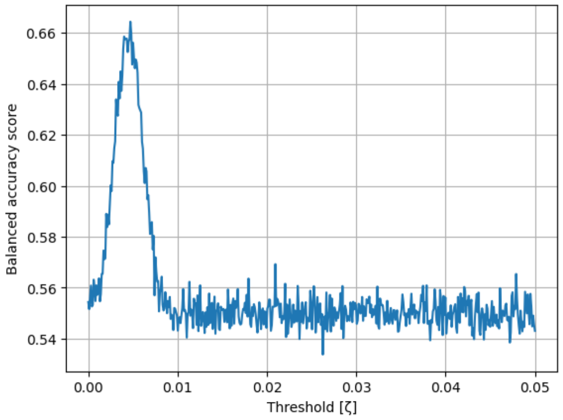

Fig. 9. Balanced Accuracy Score for Variational circuit

C. Learned Parameters

Following list summarizes the parameters found for all the architectures:

- **QVR**
  - Maximum Validation Accuracy: 0.69
  - Optimum ζ : 0.0046
- **QVR With Neural Network**
  - Maximum Validation Accuracy: 0.685
  - Optimum ζ : 0.0045
- **Variational circuit**
  - Maximum Validation Accuracy: 0.66
  - Optimum ζ : 0.00445

D. Testing

A time series(unseen by the model) with 1,000 time points was used for testing purposes. This time series was fragmented into 100 smaller time series, each containing 10 time points.There are 20 anomalous and 80 non-anomalous time series. For a comparative study, a standard threshold detection method was first applied to the testing dataset to determine if the previously described models offered any significant advantages over this commonly used method. The following list presents the testing results:

- **Threshold Detection:**
  - Balanced accuracy score: 0.61
  - Confusion matrix

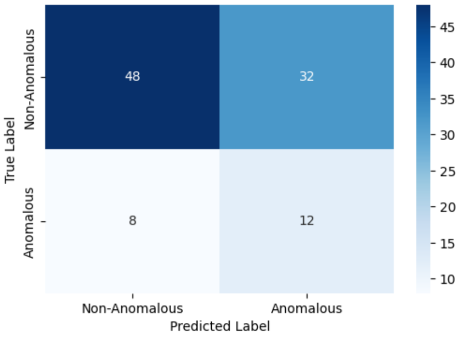

Fig. 10.

- **QVR:**
  - Balanced accuracy score: 0.70
  - Confusion matrix

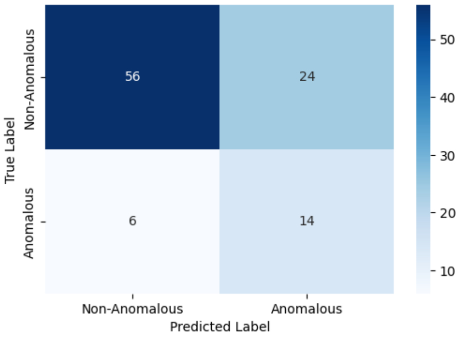

Fig. 11.

- **QVR with Neural Network:**
  - Balanced accuracy score: 0.68
  - Confusion matrix

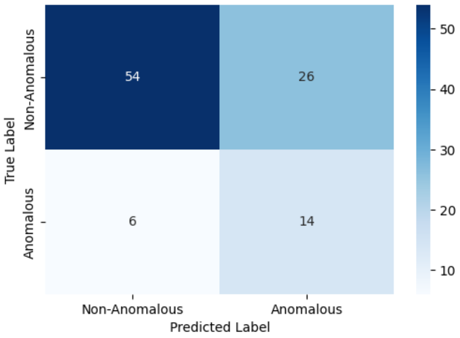

Fig. 12.

- **Variational Circuit**
  - Balanced accuracy score: 0.68
  - Confusion matrix

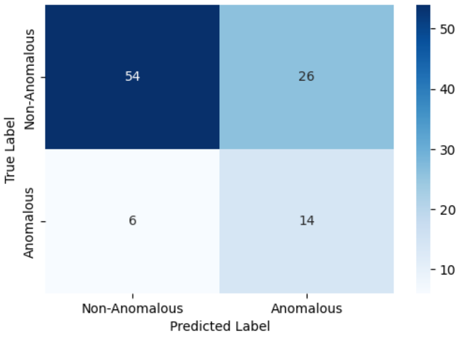

Fig. 13.

VI Observations and Conclusion

Following are the observation from the analyses:

- Based on the training performance, all the tested models, on average, produce better results than the conventional threshold scanning method, though the improvement is not very significant. The primary reason for this is likely the classical nature of the data and the relatively low system complexity, which limits the potential advantage over the threshold method. However, switching to real multi-system quantum data could result in a greater difference in accuracy between the quantum and classical models.
- No significant difference in performance was observed between the original QVR and the variational circuit using anomaly detection, with the original QVR performing slightly better. Given the much easier implementation and less resource-intensive nature of the variational circuit, it suggests that the variational circuit can be used for a variety of applications. However, for applications that require extremely precise models (such as those used in medical research, diagnosis, etc.), the conventional QVR would be more suitable, while the variational circuit could be used for other applications.
- None of these models require noise filtering in preprocessing, indicating a considerable robustness to noise in time series data. This may explain why the threshold detection method performed poorly, as it is more susceptible to noise in the signal. This susceptibility likely contributed to the high false-positive rate observed with threshold detection[Fig 10].
- The probabilistic nature embedded in QVR[1] can be effectively replaced be a neural network without affecting the architecture's performance much. This though increases the training time of the algorithm due to an increase in the number parameters.

In summary the architecture of QVR [1] can be modified in several ways in order to improve the accuracy over quantum and classical data. The work has explored some of them, showing that a simplified model (variational circuit) can have comparable results if not better. These architectures are have shown significant robustness to noise which conventional classical architectures deployed for TAD are susceptible to.

VII Future Work

Further in the research work the efficient initialization of weights and parameters[4] can contribute to improve in the accuracy of the architectures as has been demonstrated in case of classical Neural Networks in [4]. Rather than classical data data derived from actual quantum sources(Quantum sensors, Quantum simulations etc.) can be tested with the architectures. All of the quantum circuit have been simulated by using **Pennylane**, and online available device simulators by IBM. Further the execution of the architecture on present NISQ hardware with noise quantum noise mitigation techniques can be explored.

VIII ACKNOWLEDGMENT

I acknowledge the use of IBM Quantum services for this work and of NASA CMB background. from where all the data has been collected.

References

[1] Jack S Baker, Haim Horowitz, Santosh Kumar Radha, Stenio Fernandes, Colin Jones, Noorain Noorani, Vladimir Skavysh, Philippe Lamontangne, and Barry C Sanders. Quantum variational rewinding for time series anomaly detection. *arXiv preprint arXiv:2210.16438*, 2022.

[2] Johannes Bausch. Recurrent quantum neural networks. *Advances in neural information processing systems*, 33:1368–1379, 2020.

[3] Jarrod R McClean, Sergio Boixo, Vadim N Smelyanskiy, Ryan Babbush, and Hartmut Neven. Barren plateaus in quantum neural network training landscapes. *Nature communications*, 9(1):4812, 2018.

[4] Meenal V Narkhede, Prashant P Bartakke, and Mukul S Sutaone. A review on weight initialization strategies for neural networks. *Artificial intelligence review*, 55(1):291–322, 2022.