Applying Quantum Autoencoders for Time Series Anomaly Detection

Robin Frehner and Kurt Stockinger

Group 2: Buland, Lucas, Lisa, Nayeon

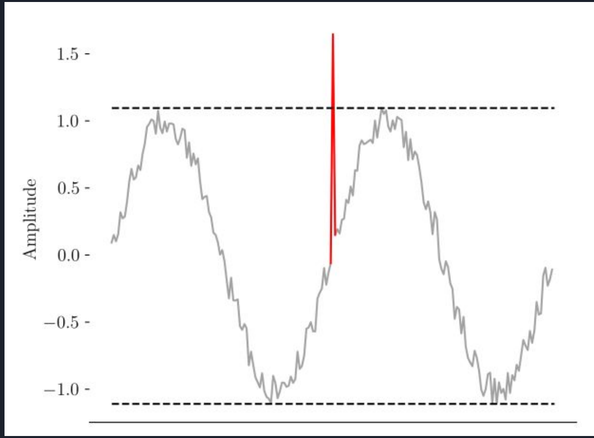
Outline

- Background Theory
- Software Implementation
- Ease-of-Reproducibility and Reproducibility Issues
- Live Demo
- Results

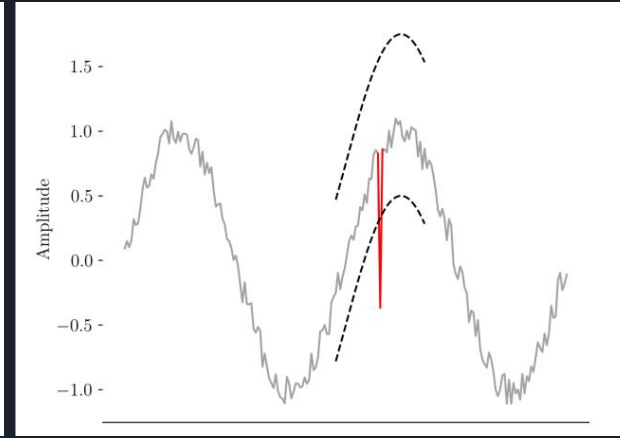
Background Theory

What is an Anomaly?

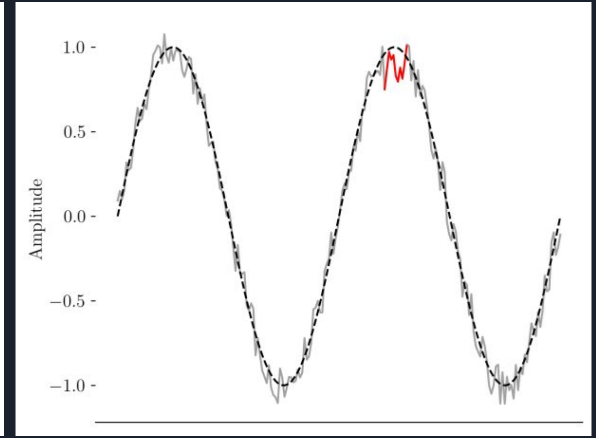
An anomaly is a data point or pattern that deviates significantly from the expected behavior.



Point Anomaly
single data point that deviates sharply from the surrounding values



Contextual Anomaly
a data point that is anomalous within its specific context



Collective Anomaly
a group of data points that together form an anomaly

What is an autoencoder?

Simple terms: It is a machine that compresses something, storing only the essential details, and then tries to expand it back out as closely to the original as possible.

It basically half-destroys a drawing and then tries to redraw it as closely to the original.

How does autoencoder help with anomaly detection?

If we train the autoencoder on data without anomalies in it,

Then we apply the trained autoencoder to data with anomalies in it,

It will have a hard time “redrawing” the parts with anomalies.

How does autoencoder help with anomaly detection?

The parts where the autoencoder had trouble tends to be really obvious (reconstruction error)

High reconstruction error = Anomaly

Motivation for using **quantum autoencoders** vs **classical autoencoders**

Quantum autoencoders is a largely unexplored field

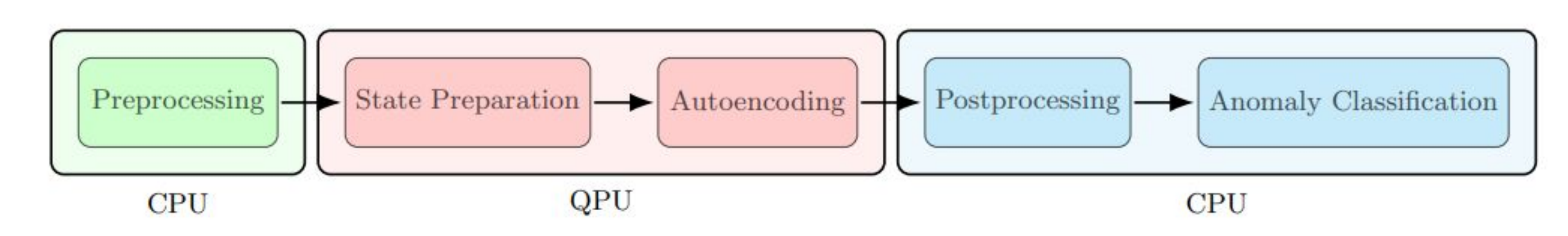
Classical autoencoders still have limitations (they struggle with data that's really large and also sometimes struggle with rare anomalies)

We want to see if quantum autoencoders can be better than classical autoencoders

Software Implementation

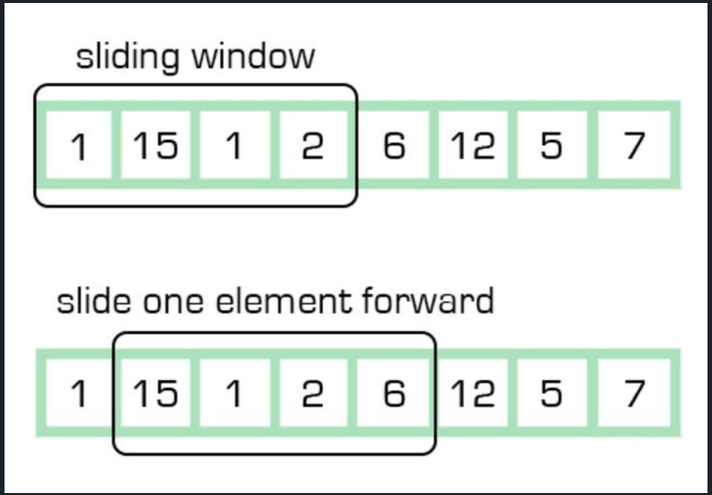
Methodology

1. Preprocessing
2. State Preparation
3. Autoencoding
4. Post Processing
5. Classification



Preprocessing

- Convert data into sliding windows of fixed size 2^K
  - K = number of qubits
- No normalization applied at this stage
  - Normalization is done during quantum state preparation



Sliding Window Technique [2]

State Preparation

- To convert each window to a quantum state, we use Amplitude Encoding
- We chose Amplitude Encoding because it requires only log(N) qubits for N data points.
- The process is deterministic and does not involve trainable parameters.

Given a window:

$$\vec{w}_i = [w_{i0}, w_{i1}, \ldots, w_{i(2^K-1)}]$$

The amplitude-encoded quantum state is:

$$|\phi_i\rangle = \frac{1}{\langle \vec{w}_i | \vec{w}_i \rangle} \vec{w}_i$$

Autoencoding

- The autoencoder uses a quantum circuit that involves an encoder and a decoder phase
- This reduces the dimensionality of the data from 7 qubits to 6 qubits + a “trash qubit” while attempting to retain essential information.

Essentially: The encoder compresses the data into a lower-dimensional quantum state, and the decoder attempts to reconstruct the original input from this compressed form.

This results in “reconstruction error”, which we will use later to detect anomalies

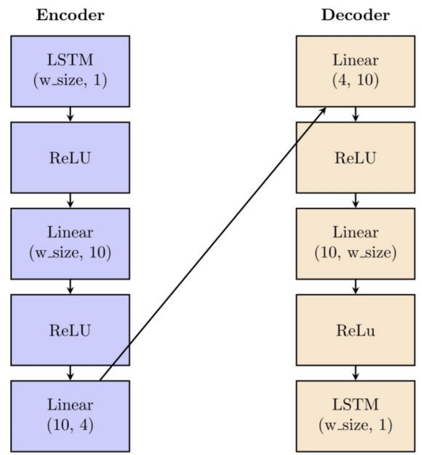
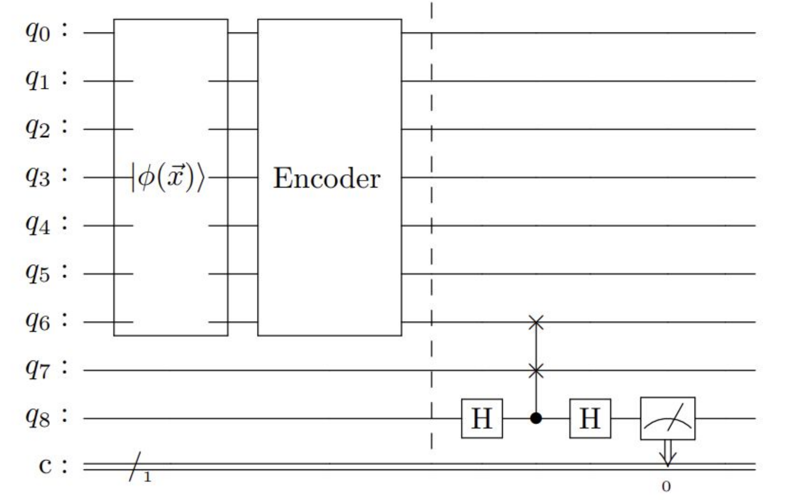
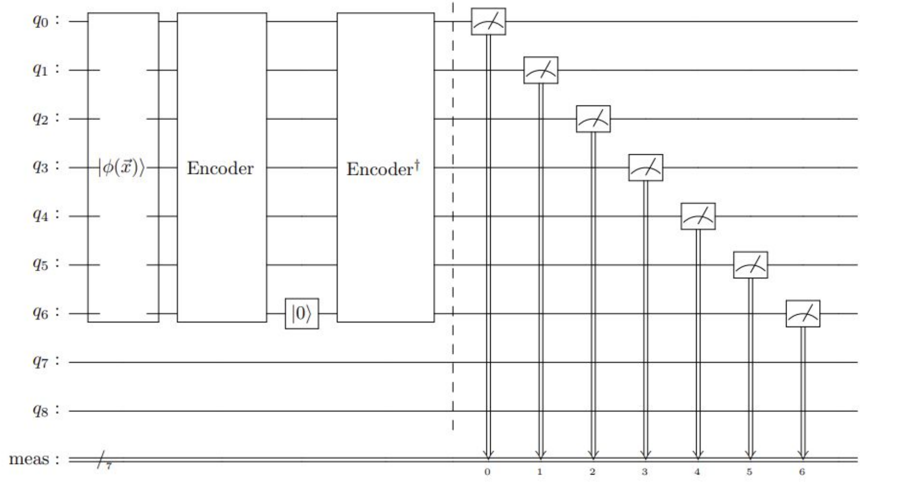


Fig. 5: Illustration of the classical deep learning autoencoder used as the baseline in this study. This architecture follows the design employed by [5]. In this work, the window size is set to 128, while other parameters remain unchanged.

Autoencoding



(a) Trainable quantum autoencoder architecture



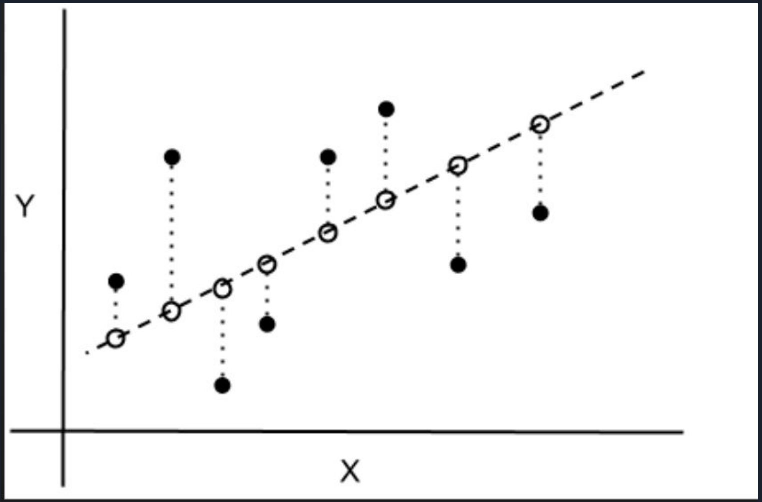
(b) Trained quantum autoencoder architecture

Trainable Autoencoder (No Decoder) - Swap Test

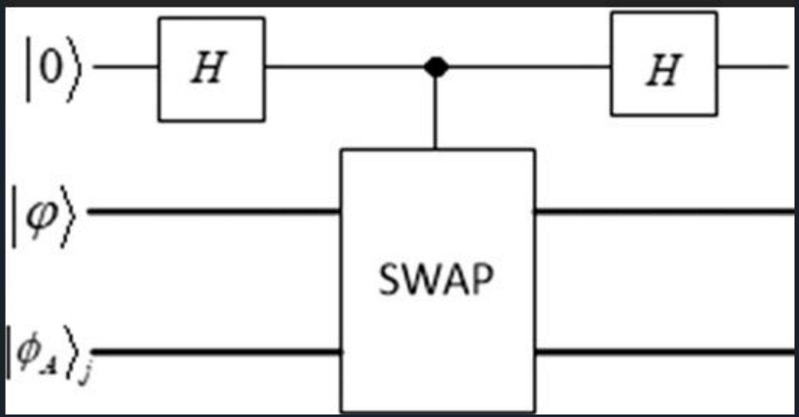
Trained Autoencoder (Decoder Used) - MSE

Post Processing i.e. Anomaly Detection

- Two methods: Mean squared error (MSE) and Swap Test



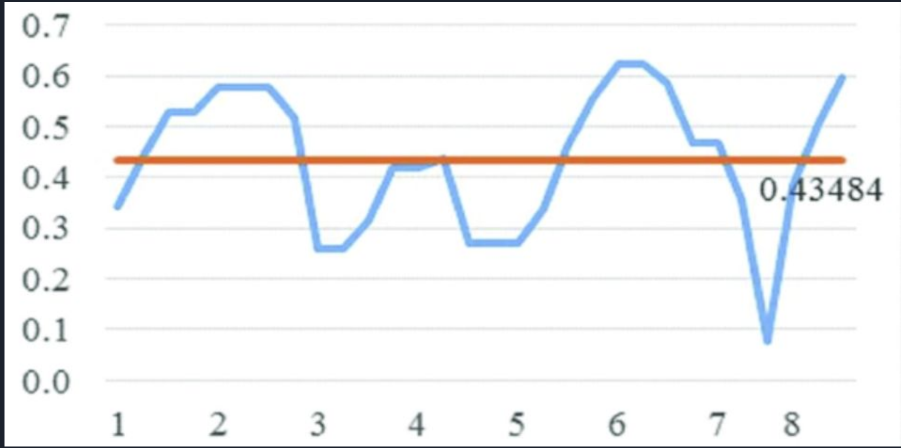
MSE



Swap Test: Actually does not rely on reconstruction error (no decoder).

Classification

- A threshold value is set on the post processed data
- Points that are consistently above the threshold are flagged as anomalies
- Great, anomaly detection is done



Ease-of-reproducibility and reproducibility issues

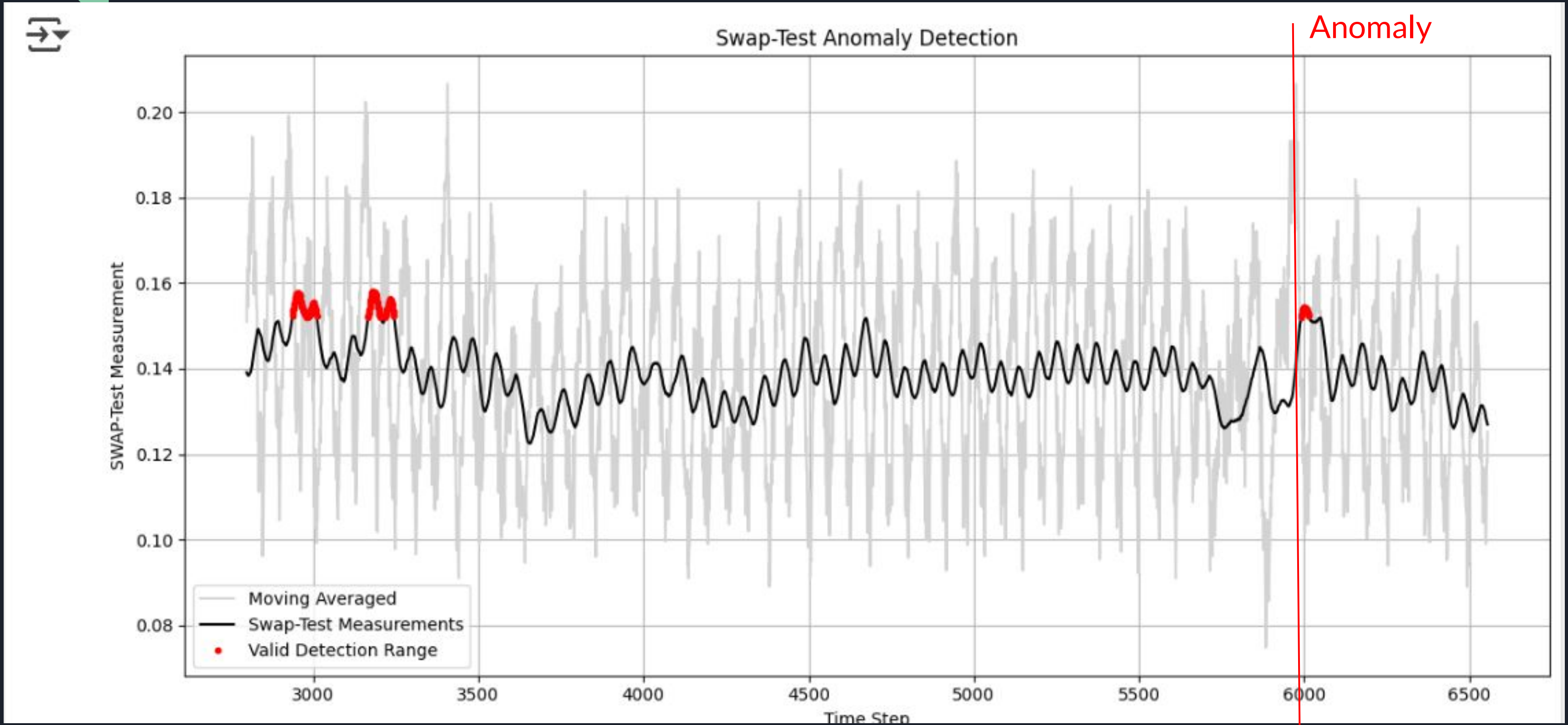
Ease of Reproduction

- Accessible and well-documented dataset
  - The dataset used is publicly available, diverse, and accompanied by thorough documentation.
- Clear methodological explanation
  - The training procedure, model setup, and evaluation steps are described in a detailed and understandable way.
- Detailed ansatz descriptions
  - Each ansatz architecture is clearly outlined, making implementation straightforward.
  - The PauliTwoDesign ansatz used a specific random seed (42) for generating their circuit.

Limitations of Reproduction

- Computational limitations - *the paper involves evaluating 5 types of ansatz, each with 9 parameters, using two different procedures (Swap and MSE), resulting in significant computational overhead.*
- Differences in the underlying gate sets or default parameterization - *differences in parameterization between Qiskit's and PennyLane's COBYLA implementations.*
- Use of separate methods (Swap and MSE) introduces complexity and increases the burden of replication.

Reproduced Result Example

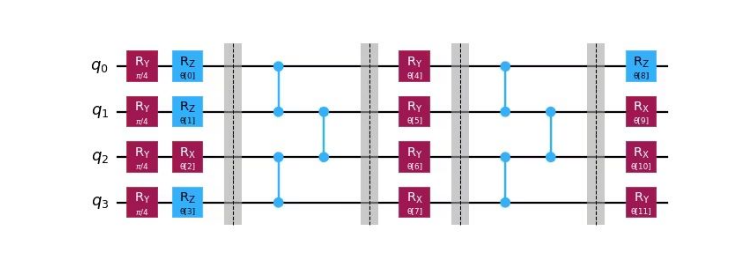


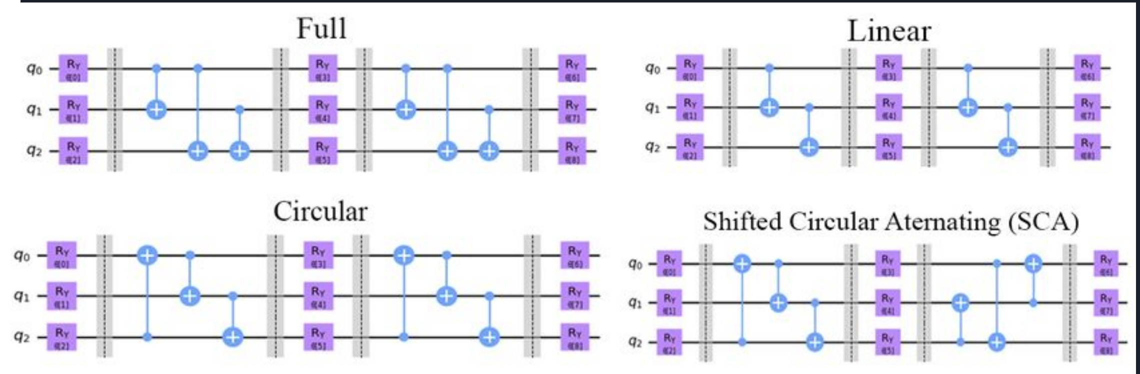
Experimental Setup

- Data set: the University of California at Riverside Time Series Anomaly Archive
- Quantum autoencoder architectures:

  PauliTwoDesign ,

  RealAmplitudes (Entanglements: Circular, Full, Linear, SCA)





Live Demo

Results

Results conclusions

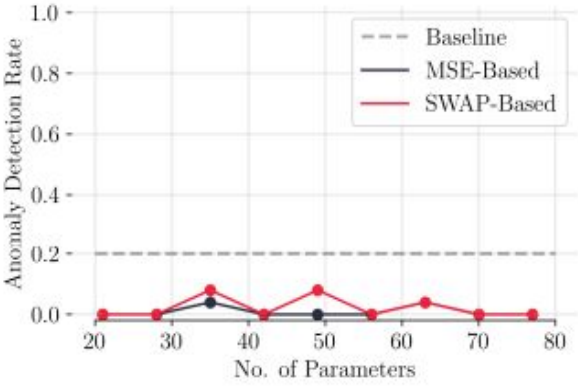
- **Quantum autoencoders outperformed classical autoencoders** for time series anomaly detection across multiple datasets.
- This superior performance was achieved with **60-230 times fewer parameters and 5 times fewer training iterations.**
- Performance varied across datasets
- Swap-Test-based anomaly detection **showed robustness at higher loss levels**.
- MSE-based detection was more effective at lower loss levels.
- For some datasets (e.g., 118, 176), Swap-Test often outperformed MSE-based detection, especially when reconstruction was challenging.
- Training loss generally increased with the number of trainable parameters for RealAmplitudes ansätze, except for RealAmplitudes-SCA at 35 parameters.
- The PauliTwoDesign ansatz showed relatively constant training loss with increasing complexity.

Results Data

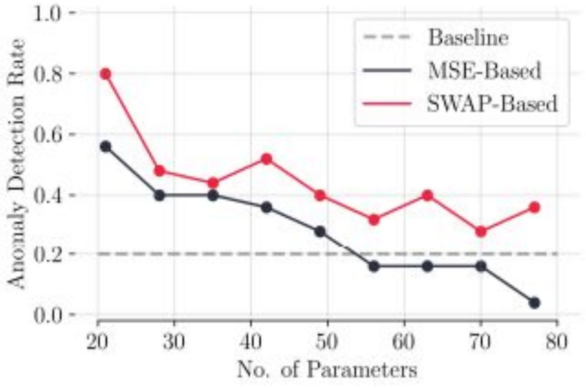
no.																			
28	PauliTwoDesign	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	RealAmplitude - Circular	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	RealAmplitude - Full	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	RealAmplitude - Linear	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
	RealAmplitude - SCA	0.00	0.00	0.20	0.00	0.00	0.00	0.20	0.00	0.00	0.00	0.00	**0.40**	0.00	**0.40**	0.00	0.20	0.00	0.00
	Classical Autoencoder					0.20									0.20				
54	PauliTwoDesign	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**0.80**	**0.80**	**0.80**	0.20	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**	**1.00**
	RealAmplitude - Circular	0.20	0.00	0.20	0.00	0.00	0.00	0.00	0.00	0.00	0.20	0.00	0.20	0.00	0.20	0.00	0.00	0.00	0.00
	RealAmplitude - Full	0.00	0.00	0.20	0.00	0.20	0.00	0.00	0.00	0.00	**1.00**	0.00	0.20	**0.40**	**0.40**	0.20	**0.60**	**0.40**	**0.60**
	RealAmplitude - Linear	**0.80**	**0.80**	**0.60**	**0.60**	0.20	0.00	0.00	0.00	0.00	**1.00**	**0.80**	**0.80**	**0.60**	0.20	**0.40**	**0.40**	0.00	0.20
	RealAmplitude - SCA	**0.80**	0.20	0.00	0.20	0.00	0.00	0.00	0.00	0.00	**0.80**	**0.60**	0.00	**0.60**	0.20	0.00	0.00	0.00	0.00
	Classical Autoencoder					0.20									0.20				
99	PauliTwoDesign	0.00	0.00	0.00	0.00	0.00	**0.40**	0.00	0.00	0.00	0.00	0.00	0.20	0.00	0.00	**0.40**	0.00	0.00	0.00
	RealAmplitude - Circular	0.00	0.00	0.00	0.20	0.00	0.00	0.00	0.00	0.20	0.00	0.00	0.00	0.20	0.00	0.00	0.00	0.20	**0.40**
	RealAmplitude - Full	0.00	0.00	0.00	0.20	0.20	0.00	0.20	0.00	**0.40**	0.20	0.00	0.00	0.20	0.20	0.00	0.00	0.20	0.20
	RealAmplitude - Linear	0.00	0.20	0.00	0.00	0.20	0.20	0.20	0.00	0.20	0.00	0.00	0.00	0.00	0.20	0.20	**0.40**	0.20	0.20
	RealAmplitude - SCA	0.00	0.00	0.00	0.00	0.20	0.00	0.00	**0.40**	0.20	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.20
	Classical Autoencoder					0.20									0.20				
118	PauliTwoDesign	0.00	0.00	0.00	0.00	0.00	0.20	0.00	**0.20**	0.00	**0.40**	**0.40**	**0.40**	**0.60**	**0.60**	**0.60**	0.20	**1.00**	**0.60**
	RealAmplitude - Circular	0.20	**0.40**	0.00	0.20	**0.40**	0.00	0.20	0.00	0.20	**0.80**	**0.80**	**0.80**	**0.60**	**1.00**	**1.00**	**0.80**	**0.60**	**1.00**
	RealAmplitude - Full	0.00	**0.80**	**0.60**	0.20	**0.40**	**0.40**	0.20	**0.40**	0.20	**0.80**	**1.00**	**1.00**	**0.80**	**0.40**	**0.80**	**0.60**	**0.80**	**0.80**
	RealAmplitude - Linear	0.00	0.00	0.00	0.20	0.40	0.20	**0.80**	0.20	0.20	0.20	**0.60**	0.20	**0.60**	**0.80**	**0.60**	**0.80**	**0.80**	**0.80**
	RealAmplitude - SCA	0.00	0.20	**0.60**	0.20	0.00	0.20	0.20	0.00	0.00	**0.60**	**0.60**	**0.60**	**0.80**	0.00	**0.80**	**0.80**	**0.60**	**1.00**
	Classical Autoencoder					0.20									0.20				
138	PauliTwoDesign	0.00	0.00	0.00	**0.40**	**0.40**	0.20	**0.40**	0.00	0.00	0.20	0.00	0.00	**0.40**	**0.80**	0.20	0.00	0.20	0.00
	RealAmplitude - Circular	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.20	0.00	**0.40**	0.20	0.00	0.00	0.20
	RealAmplitude - Full	0.00	0.00	0.00	0.20	0.20	0.00	0.00	0.00	0.20	0.20	0.00	0.00	**0.40**	0.20	0.00	0.20	0.00	0.20
	RealAmplitude - Linear	0.00	0.20	0.00	0.00	0.00	0.00	0.00	0.00	0.20	0.20	0.20	0.00	0.00	0.00	**0.40**	0.20	0.20	0.00
	RealAmplitude - SCA	0.00	0.00	**0.60**	0.00	0.20	0.20	**0.40**	0.00	0.00	**0.40**	0.00	**0.60**	0.20	**0.60**	**0.40**	**0.60**	0.20	**0.40**

Anomaly Detection and Model Characteristics

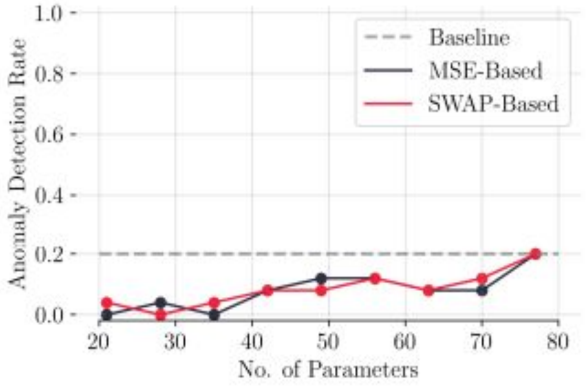
- MSE requires low train loss to be effective otherwise use Swap-based methodology
- Swap-based methodology can be superior as it does not rely on accurate reconstruction
- PauliTwoDesign ansatz showed near-constant training performance across different complexities
- The relationship between model complexity (number of parameters) and training loss is not always straightforward . For RealAmplitudes ansätze, training loss tended to increase with the number of parameters.
- The **PauliTwoDesign ansatz showed near-constant training performance** across different complexities



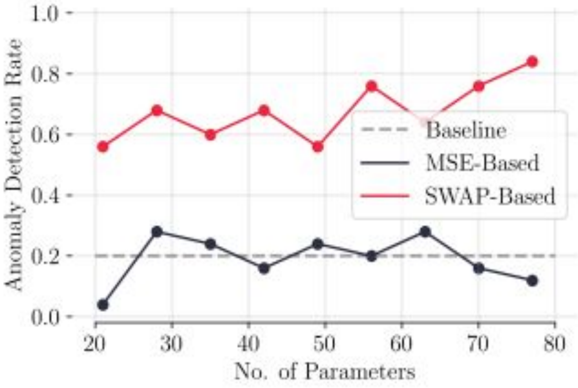
(a) Dataset no. 28



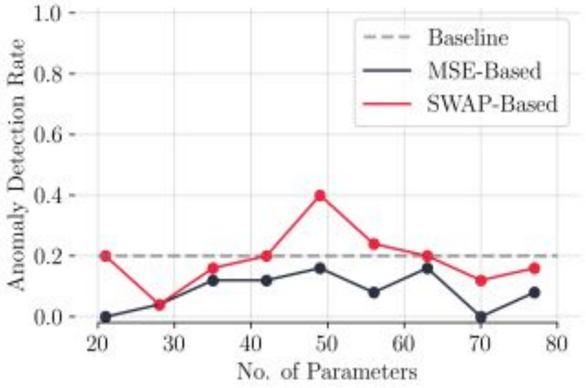
(b) Dataset no. 54



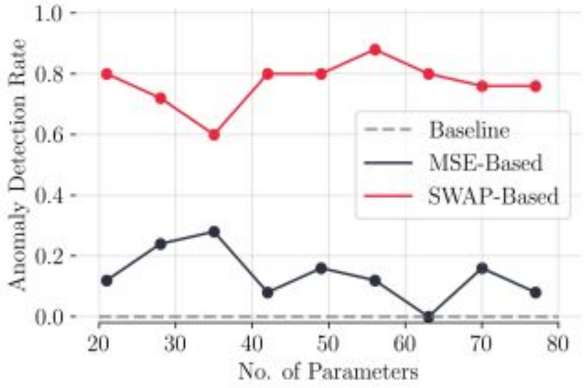
(c) Dataset no. 99



(d) Dataset no. 118



(e) Dataset no. 138



(f) Dataset no. 176

Issue to Approach

- Direct amplitude encoding on real NISQ hardware encounters significant limitations due to extensive hardware requirements
- Experiments on real quantum hardware showed higher training loss compared to simulations
- **Training dynamics and final model parameters obtained in simulations do not directly translate to the real hardware scenario**
- Computationally intractable for larger qubit counts due to the exponential growth of the state vector size
- **21 hours of quantum computing time**
- **Cost of 120,960 USD for the paper on real hardware**

Thank You

References

- [1] Applying Quantum Autoencoders for Time Series Anomaly Detection by Robin Frehner and Kurt Stockinger, source- https://arxiv.org/pdf/2410.04154v2
- [2] Image source: https://medium.com/@yingpeng0221/sliding-window-algorithms-a-comprehensive-introduction-71f89b6db4ff