Design Considerations for Denoising Quantum Time Series Autoencoder

Jacob L. Cybulski[1][0000−0002−9061−9389] and
Sebastian Zając[2][0000−0001−6616−8744]

[1] Deakin University, Melbourne, Australia
jacob.cybulski@deakin.edu.au
https://jacobcybulski.com/
[2] SGH Warsaw School of Economics, Warsaw, Poland
szajac2@sgh.waw.pl
https://sebastianzajac.pl/

Abstract. This paper explains the main design decisions in the development of variational quantum time series models and denoising quantum time series autoencoders. Although we cover a specific type of quantum model, the problems and solutions are generally applicable to many other methods of time series analysis. The paper highlights the benefits and weaknesses of alternative approaches to designing a model, its data encoding and decoding, ansatz and its parameters, measurements and their interpretation, and quantum model optimization. Practical issues in training and execution of quantum time series models on simulators, including those that are CPU and GPU based, as well as their deployment on quantum machines, are also explored. All experimental results are evaluated, and the final recommendations are provided for the developers of quantum models focused on time series analysis.

Keywords: Quantum machine learning · Autoencoder · Quantum encoding · Quantum measurement.

1 Introduction

The area of quantum time series analysis finds its foundations in a mature and well-published field of classical time series analysis [14], as well as in the new, dynamic, and not yet tested area of quantum machine learning [27].

Our exploration of quantum machine learning approaches to time series analysis has been motivated by our long-term interest in quantum methods for the detection and removal of irregularities in temporal data [4].

In general, there are two categories of irregularities found in time series [4]: (1) noise and errors, or unwanted data that need to be removed from data; and (2) anomalies, representing events of interest, which need further analysis and understanding, often with great urgency.

ICCS Camera Ready Version 2024
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-63778-0_18

The presence of abnormal data in time series may be due to poor data entry practices, the use of substandard recording devices, or the impacts of environmental factors. Eliminating noise could improve the quality of analytical models trained with these data [1]. For example, the removal of noise and errors from time series could improve the accuracy of systems responsible for financial forecasting [28], monitoring the condition of industrial machines [17], or helping physicians perform medical diagnoses [30]. However, identification and understanding of anomalies discovered in financial transactions could reveal fraudulent customer behavior, anomalies in machine vibration could represent the onset of its catastrophic failure, while anomalies in ECG recordings could signify early signs of heart attacks.

The handling of simple noise (such as Gaussian) and simple anomalies (such as jumps in amplitude or shifts in time) can be handled by using simple filters and algorithmic solutions [19]. More complex temporal patterns involve the development of machine learning models [20]. In more demanding cases, methods to eradicate time series problems require massive computing resources, leading researchers to explore quantum solutions [25,15].

Much of this area of research is exploratory and includes a lot of fragmented and highly specialized "proof-of-concept" projects. These explorations include quantum signal processing [11], evaluating similarity measures to support temporal data analysis [21], methods of quantum time series classification [32], quantum time series forecasting [12], the adoption of deep learning models, such as RNN and LSTM, for quantum modeling [2,29,8], and some work on noise removal and anomaly detection in time series [18].

The machine learning model adopted in this project aims to analyze time series using a neural network autoencoder (AE) [13, ch14], and more specifically its quantum counterpart, the quantum autoencoder (QAE). In general, autoencoders feature lossy data compression [10], allowing data sequences to preserve their most important or recurrent features, while removing their less significant, noisy, or infrequently occurring data [9, sect. 2.2.2].

Although there are many applications of classical AEs [16,3], there are very few examples of QAE use. The best received QAE studies include efficient data compression [25], image processing [15,5], analysis of marketing media [24], and anomaly detection in signals, although only in the frequency domain [26].

Other research explored the ability of QAE to deal with data in the time domain. They investigated the impact of data encoding on QAE performance [5], performed data sequence compression and reconstruction with high accuracy [25], and efficiently managed resources in hybrid quantum-classical computational settings [15,26].

Applications of quantum autoencoders for time series analysis, compression of temporal data, noise elimination, representation learning, and anomaly detection are all in uncharted territory and hence worth further investigation.

ICCS Camera Ready Version 2024
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-63778-0_18

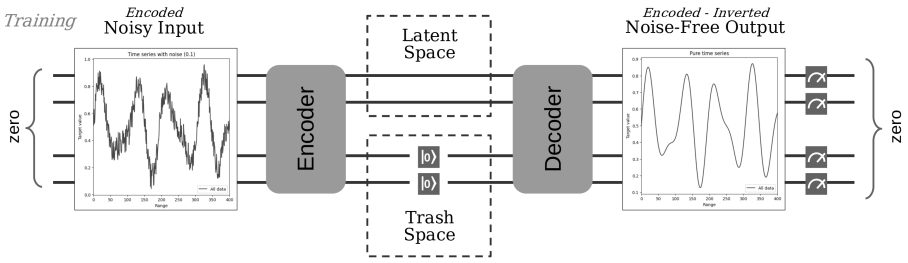


Fig. 1: Full QAE for circuit training (all qubits measured).

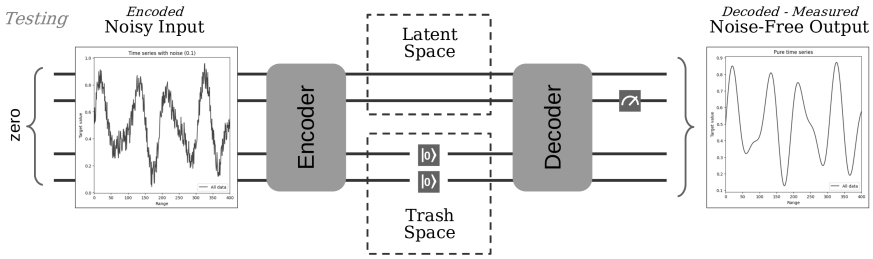


Fig. 2: Full QAE for circuit testing.

2 QuTSAE design

There have been few prior attempts at applying quantum autoencoders to time series analysis. The design of the well-known *quantum time series autoencoders* (QuTSAE) commonly utilizes the variational quantum algorithms [7], which manipulate quantum circuits with parameterised operations (or gates) (see Figures 1 and 2). QuTSAE operations are arranged into blocks that include the following:

- *Input block* coding classical data as the state of a quantum circuit.
- *Trainable encoder block* compressing the quantum state into a latent space.
- *Latent space* defining an active subset of qubits and discarding (reinitializing) the state of the remaining "trash" qubits.
- *Trainable decoder block* decompressing the quantum state of the latent space.
- *Output block* reversing the circuit state to the initial state (zero).
- *Measurements* converting the final quantum state into classical data.

2.1 Design choices for QuTSAE architecture

When designing the QuTSAE model, the following well-known architectural choices have been considered.

Replicating QAE
Initially, consider the QAE training model (see Figure 1), of which the latent space includes all qubits, the output block inverts the gate sequence of the input block, and a decoder is the inverse of an encoder with both sharing

ICCS Camera Ready Version 2024
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-63778-0_18

their parameters. The training circuit works as an identity that reverses the quantum state around the latent space, to reproduce the initial quantum state $|0\rangle^n$, regardless of the parameter values. As the QAE testing model (see Figure 2) drops the training circuit's output block, its decoder returns a quantum state encoded on input. The final state can be measured in a variety of ways - all qubits at once or one at a time (as shown in the figure).

Approximating QAE

When the size of the latent space is reduced (see Figure 1), the training QAE could no longer act as an identity for all sets of parameter values. However, the QAE encoder/decoder parameters can be optimized to minimize error in measured circuit values. When the output block is removed (see Figure 2), the QAE circuit returns an approximation of the input state on the output. The smaller the size of the latent space, the more information is lost by the circuit and the more imprecise the reconstruction of the input on the output.

Denoising QAE

Autoencoders are known for their ability to denoise data. However, since the input/output and encoder/decoder blocks are their respective mirror twins, when QAE is given some noisy input, the same noise would be reconstructed on output. To prevent this from happening, the input/output and encoder/decoder pairs need to be decoupled by making their parameter sets distinct. In this case, in QAE training (see Figure 1), input and output can be assigned noisy and pure data, respectively, and encoder / decoder parameters can be optimized independently, thus allowing the reconstruction of pure data from noisy data.

Custom QAE

Other types of QAEs have been developed, and in particular hybrid neural networks, with some interface (called dressing) between classical and quantum components [24].

2.2 Design choices for QuTSAE input encoding

As the above QAEs make no assumptions about data on input and output, it is possible to apply them to time series and implement a functional QuTSAE.

Time series have to be represented in a way that allows the model to continuously fit the series and its subsequences. A sliding-window protocol was therefore selected to represent and process time series subsequences. The time series was also preprocessed by differencing and scaling its values, making the series (partially) stationary and its values manageable (as per ARIMA), [14, ch 9.1, 13.3].

As the window moves along the series during training or testing, its values need to be encoded as the QuTSAE quantum states of the input and output blocks. It is preferable to adopt a single encoding scheme for these two purposes.

There are several ways of quantum encoding schemes that are applicable to time series, i.e., basis, amplitude, QRAM, angle, and others [31]. Each of these schemes has its own strengths and weaknesses. For example:

ICCS Camera Ready Version 2024
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-63778-0_18

Basis encoding
adopts the binary representation of numbers on input; it is simple, but limits the circuit to handle single values only or multiples of imprecise values.

Amplitude encoding
represents a window as a distribution of expectation values, as in circuit measurement; great for results interpretation; however, encoding of different input values leads to circuits of different structures, making very inefficient execution on QPUs or GPUs.

QRAM encoding
precodes all window (or time series) values in the circuit and allows "referring" to them as needed; very flexible in processing; however, leads to large circuits, making their execution on quantum devices prone to errors.

Angle encoding
encodes time series values as qubit state rotations, making it flexible, efficient, circuit size friendly, and easy to manipulate and measure. Its main weakness is its susceptibility to quantum noise, which could affect the accuracy of the results obtained from noisy NISQ-term quantum machines.

Angle encoding, which was adopted for QuTSAE (see Figure 3), can be realized as a series of $(Ry(x_0), Ry(x_1), ..., Ry(x_n))$ qubit rotations around the y axis. To facilitate later measurement and interpretation of the encoded values, we selected the state $|+\rangle$ to represent the value 0, $|1\rangle$ as -1, and $|0\rangle$ as $+1$. Any adjustment in a qubit $value \in [-1, +1]$ is a Ry rotation of $value \times 0.5\pi \times (1-2\times\epsilon)$ (with ϵ as an error range). Due to time series differencing, the values handled by QuTSAE are small, resulting in qubit states very close to $|+\rangle$.

It is worth mentioning that early experiments with the Z and ZZ feature maps, which are commonly used to encode input data, were found to be difficult to use along the QAE decoder for output reversal and result interpretation.

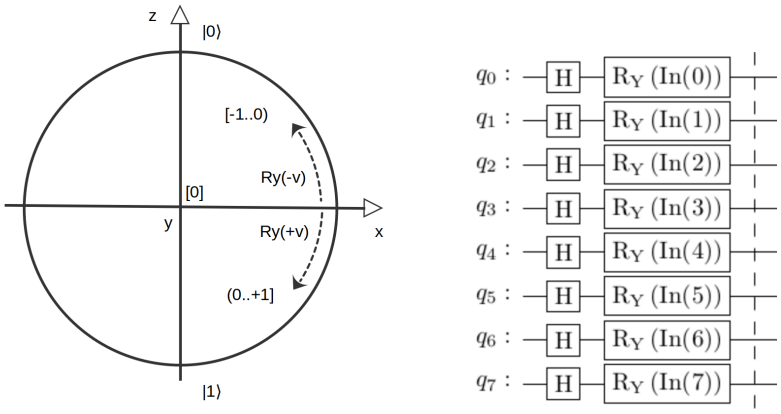
2.3 Design choices for QuTSAE output decoding / cost function

Measurement and interpretation of the QuTSAE model are vital to the successful optimization of its parameters through a cost function; and, to gaining the ability to extract the reconstructed input from the output block.

Half-QAE with swap test
When dealing with approximating QAEs, where the input-encoder pair shares its parameters with the decoder-output pair, the QAE circuit can be trained on the input-encoder half of the circuit alone [25].

This can be accomplished with a cost function using a *swap test* circuit, which aims for the measurement of the trash state to approach zero, and thus for the state of the latent space to represent the maximum of information present on input. Unfortunately, when the trash size is large (say, greater than 5 qubits), the swap test becomes slow and inefficient.

ICCS Camera Ready Version 2024
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-63778-0_18



(a) TS value angle encoding. (b) TS input encoding circuit.

Fig. 3: QuTSAE input encoding

Full-QAE and interpretation of measurements

When data are highly redundant, the model's latent space can encode input without any loss of information with just a few qubits, in which case the swap test is less effective working with large trash area.

Instead, in model training, the optimizer could measure the state of the entire input-encoder-latent-decoder-output sequence, which should approach the initial state of zero, when the probability of all qubits to be simultaneously zero is $p(|0\rangle^n) \approx 1$ (n is the number of qubits, see Figure 1). This is achieved by minimizing the cost function of the form $Cost = 1 - p(|0\rangle^n)$, whose implementation requires measuring all qubits at once and testing a single distribution outcome of $|0\rangle^n$.

In model testing (with the output block removed), the output value can be derived from the angular state of individual qubits. This can be calculated from the probability distribution of measuring all qubits at once. However, as the number of qubits increases, the space of expectation values grows exponentially, resulting in the calculation to be of exponential complexity. To overcome this problem, it is possible to measure every qubit state individually (see Figure 2 as an example), which provides single-qubit expectation values, easily interpreted as qubit amplitudes, cast back to qubit Ry rotations, and decoded as part of the time series output. Although the latter process has linear complexity, with the small number of qubits and a long execution time of quantum simulation, in practice the process is much slower.

2.4 Design choices for QuTSAE encoder/decoder ansatze

Another factor that affects the QuTSAE model is the selection of an *ansatz* implementing the QAE encoder and decoder. An anzatz is a template with

ICCS Camera Ready Version 2024
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-63778-0_18

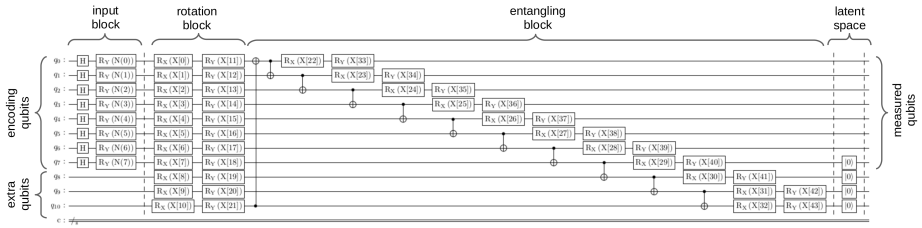


Fig. 4: QAE input block, encoder (with 3 extra qubits, Rx and Ry gates, and 1 rotation/entangling block) and a latent space. A decoder and an output block (for training) are not shown (an inverse of the input and encoder structure).

trainable parameters that can be assigned specific values to create an executable circuit instance. A great variety of anzatz types are available from quantum libraries (such as Qiskit or PennyLane). They can be customized to have specific structural characteristics, the type and number of rotation and entanglement blocks, and other properties (see Figure 4).

The following are four important aspects of the QuTSAE ansatz design.

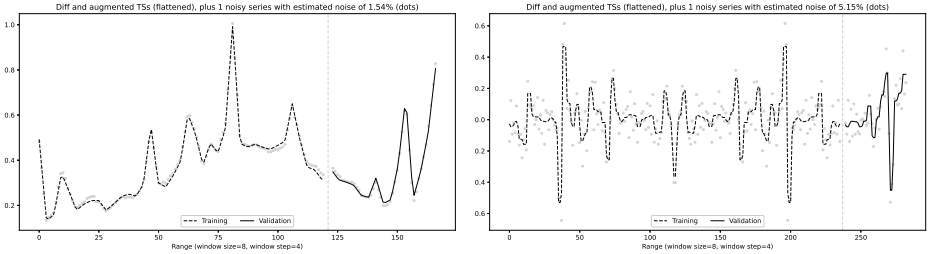
Ansatz structural symmetry
The structural symmetry between the encoder and decoder ansatze is necessary to ensure the reversibility of the QAE components on both sides of the latent space and the ensuing effectiveness of the adopted cost function. This decision led to the rejection of ansatz structures that interweave input and encoding blocks to facilitate reuploading of input data [23]. While data reuploading improves circuit trainability, it breaks the QAE symmetry and prevents easy interpretation of the output.

Ansatz rotation blocks
To keep the structure of the QuTSAE ansatz simple and consistent with the Ry input encoding and interpretation of the output, the ansatz was initially designed using only the Ry rotations. However, this imposed a severe limitation on the possible circuit states, which impeded the model learning. Subsequently, to take advantage of the entire space of possible qubit states, rotation blocks consisting of Rx and Ry gates were adopted.

Ansatz size (width and depth)
The QuTSAE performance is influenced by the ansatz size, i.e. its width (the number of qubits) and depth (the longest gate path). In QuTSAE, the ansatz width is controlled by extra qubits that offer additional degree of freedom but do not participate directly in input/output activities (option *aw*), and its depth by the number of rotation/entangling blocks (option *reps*).

Optimization of the ansatz parameters
Typically, the model optimizer cannot be selected without performing a preliminary investigation of the model parameter space and the effectiveness of the optimization algorithm. For example, this project reviewed gradient-based optimizers (such as ADAM and SPSA), as well as linear and nonlin-

ICCS Camera Ready Version 2024
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-63778-0_18



(a) QAE original data. (b) QAE differenced data.

Fig. 5: Fragment of the original and differenced beer sales data (training set as a dashed line, validation as a solid line, noise shown as dots).

ear approximation methods (such as COBYLA and BFGS), with COBYLA found to be the quickest, most effective and producing the best results.

In our design deliberation of QuTSAE time series processing, we initially adopted the approximating half-QAE model developed by Romero et al. [25] and the enhancements recommended by Bravo-Prieto [5]. The resulting models produced a high level of accuracy in replicating inputs into outputs, with or without noise - the behavior undesirable in sequence denoising. Subsequently, the full-QAE model was proposed, leading to slower model optimization, but more successful noise reduction in time series.

3 Experiments

A series of experiments were conducted to test the ability of QuTSAE to remove "simple noise" from temporal data, and investigate the influence of model design decisions on its performance in training and validation. To manage the complexity of noise presence in the data and noise generated by quantum machines, it was decided to develop QuTSAE models using noise-free quantum simulators.

Data used for QuTSAE training related to beer sales in the USA, which was sampled from the IRI Marketing Data Set [6]. We selected a small sample of time series data consisting of 160 data points, which was split into two partitions (0.75/0.25) - 120 data points for model training and 40 data points for its validation (see Figure 5). We refer to the original data as *pure*. A copy of the pure data was injected with 3% of uniformly distributed noise, and consequently we refer to these data as *noisy*. Differencing was applied to all pure and noisy data sets, resulting in data sets with noise exceeding 5.8%.

Subsequently, all data sets were segmented into windows of 8 data points each, sliding with a step of 4. This resulted in 29 training windows and 10 validation windows. Note that the window size of 8 (and various step sizes) was established in earlier experiments with half- and full-QAEs and synthetic data. After some investigation, it was decided that a window of size 8 and a step of 4 was suitable for the experiments with full-QAEs and beer sales data.

The experiments were carried out on a Qiskit quantum simulator running on Ubuntu 22.04.4 LTS, on a workstation equipped with an i9 CPU (24 cores and 32

ICCS Camera Ready Version 2024
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-63778-0_18

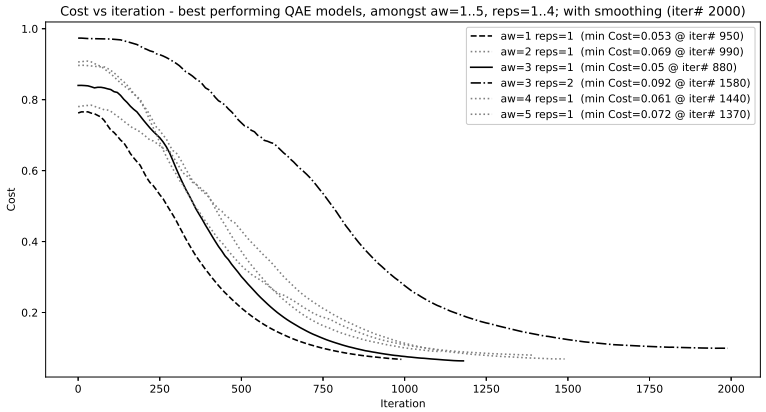


Fig. 6: Cost vs iterations in training models of the same latent space (lat=7) but different width and depth. Based on training cost, 5 best models (out of 14).

threads), 64GB RAM, and a NVIDIA GPU GeForce RTX 4090. Initially, all experiments used GPU; however, later code improvements resulted in significantly faster CPU-based runs.

3.1 Determining the optimum ansatz size

The first group of experiments aimed to decide on the optimal size of the QuTSAE ansatz. Fourteen separate denoising models were created in Qiskit using a *TwoLocal* anzatz. All models shared the same number of input/output qubits (8, same as window size), the same size of the latent space (of 7 latent qubits, plus 1 trash qubit to test the QuTSAE data compressing behavior), same type of entanglement (shifted circular-alternating "sca" entanglement), and parameter training blocks (using Rx and Ry rotations). However, the models differed in their width (aw = 1..5), rotation / entanglement block repetitions (reps = 1..4), and the resulting number of trainable parameters (varying from 64 to 180).

In training, models with narrower and shallower circuits, and consequently smaller numbers of parameters, converged the quickest (within 1000 iterations). Models with wider and deeper circuits, featuring a large number of parameters, needed more training time (up to 2000 iterations; see Figure 6).

In each optimization step (see Table 1, Ex. 1), the training cost and model parameters were saved for later analysis. The parameters were used to derive the training and validation scores R^2 (R-squared), $RMSE$, MAE, and $MAPE$ metrics for all intermediate stages of each model evolution. The $MAPE$ score was not used in the comparison of the performance of the models. It was, however, produced to provide information on the distribution and variance of the output (with respect to input).

Based on the MAE validation scores, among the models with a latent space of 7 qubits, two were selected for further experiments (see Table 1, Ex. 1 models with additional depth $aw = 3$, and repeating blocks $lat = 1$ and $lat = 2$).

ICCS Camera Ready Version 2024
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-63778-0_18

Table 1: Cost and scoring results (top 4 models in Ex. 1, 2a and 2b experiments)

Experiment			Min	Training				Validation			
Lat	Aw	Reps	Cost	R2	RMSE	MAE	MAPE	R2	RMSE	MAE	MAPE
Ex. 1 (Qiskit/QNN)											
7	3	2	0.089	0.767	0.081	0.059	3.592	0.803	0.074	0.058	1.421
7	3	1	0.054	0.742	0.085	0.067	4.471	0.782	0.078	0.064	1.946
7	5	1	0.078	0.738	0.086	0.071	3.416	0.767	0.081	0.068	2.455
7	4	1	0.065	0.658	0.097	0.074	3.880	0.723	0.088	0.072	2.191
Ex. 2a (Qiskit/QNN)											
7	3	1	0.054	0.742	0.085	0.067	4.471	0.782	0.078	0.064	1.946
6	3	1	0.070	0.399	0.130	0.087	2.745	0.442	0.116	0.088	2.367
4	3	1	0.063	0.428	0.126	0.085	3.346	0.331	0.137	0.094	1.568
5	3	1	0.066	0.455	0.123	0.095	4.603	0.403	0.129	0.097	2.654
Ex. 2b (Qiskit/QNN)											
7	3	2	0.089	0.767	0.081	0.059	3.592	0.803	0.074	0.058	1.421
8	3	2	0.086	0.761	0.083	0.066	4.743	0.741	0.085	0.068	1.952
2	3	2	0.086	0.752	0.085	0.068	4.668	0.690	0.090	0.072	2.490
5	3	2	0.105	0.396	0.132	0.088	3.122	0.461	0.115	0.079	1.473
Ex. 3 (PyTorch/MLP, lat=7)											
Enc=3/Dec=3			0.012	0.996	0.010	0.006	0.225	0.751	0.081	0.059	1.310

3.2 Impact of latent space on performance

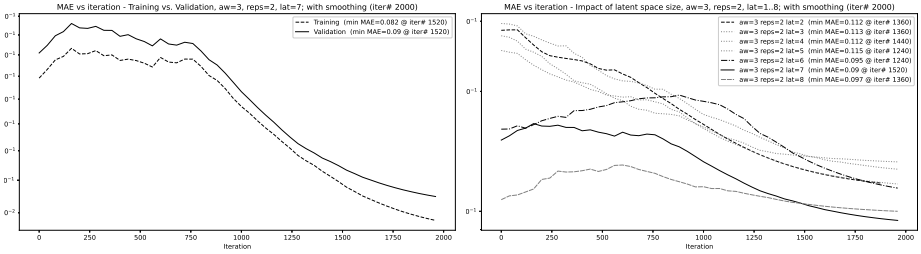
As the size of the latent space determines the quality of time series reconstruction, the second group of experiments was conducted to identify the optimum ratio between latent and trash space in the QuTSAE circuits.

We used the two models selected previously and varied the size of their latent space, from 1 to 8 qubits (with size 0, interaction between the encoder and the decoder was not possible, thus generating errors). This resulted in 14 additional models to be investigated. Each model was optimized, its training and validation scores plotted (e.g., see Figure 7a), and those with suboptimal performance were rejected. Subsequently, the models within each size group were compared (e.g see Figure 7b for models of the same size, given as $aw = 3$ and $reps = 2$).

As indicated by MAE validation scores, the best 4 models were then selected from each model size group (see Table 1, Ex. 2a and Ex. 2b). By analyzing their performance scores, the two best performing models overall were found to have a ratio of 7:1 latent to trash space (the most desirable ratio for this data set), 3 additional qubits (beyond the 8 input/output quibits), and 1 to 2 rotational/entangling blocks.

To assess the performance of QAE models against the equivalent classical autoencoders, a number of such models were developed using the PyTorch package. Each model had a differently sized latent space (from 1 to 8 variables) and consisted of two multilayer perceptrons (MLP) acting as the AE encoder and decoder, featuring 3 hidden layers each, and a total of 19,741 parameters (see Table 1, Ex. 3 PyTorch model with $lat = 7$). Training of PyTorch and quantum models used the same windowed and differenced data, and identical approaches

ICCS Camera Ready Version 2024
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-63778-0_18



(a) MAE scores for training and validation (aw=3, reps=2, lat=7, y linear scale).

(b) Latent space impact on validation with MAE scores (aw=3, reps=2, y log scale).

Fig. 7: QuTSAE model performance.

to data coding. Although PyTorch models excelled in their training performance, their validation performance was on par with that of quantum models.

3.3 Time series denoising

The final experiment was to investigate whether the QuTSAE models are capable of denoising time series.

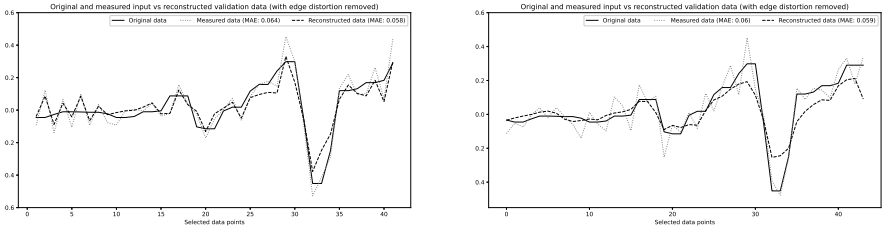
To this end, for each model developed so far, we created its two separate instances: the first by instantiating its parameters with values found to be optimal during training, and the second by instantiating its parameters with values determined to be optimal for model validation. We then re-examined all model instances by applying the previously used scoring metrics to assess differences between reconstructed vs. pure sequences, as well as differences between noisy vs. pure sequences. Should the reconstructions be closer to the pure sequences than noise, we would then regard such models as capable of denoising time series.

Significantly, the final model scoring was carried out in two stages. As the time series windows had some significant overlaps due to their stepwise creation, in the first scoring process, we identified data points of high score variance commonly present at window edges, which were removed to improve the model scores produced in the second stage. We then reintegrated the remaining window overlaps by averaging to produce the adjusted sequences suitable for the presentation of the pure, noisy and reconstructed series (see Figure 8).

As an example, let us take the best performing QuTSAE model, characterized by its hyperparameters lat = 7, aw = 3, and reps = 2. Table 2 provides its training and validation scores for pairs of pure, noisy, and reconstructed sequences. In the left column, we find the scores comparing the original pure time series vs. the measurement of noisy time series. In the right column, the scores compare pure time series vs. the reconstruction of pure time series from noisy input.

In validation, the scores of $R2$, $RMSE$, and MAE indicate that the reconstructed sequences fall between pure and noisy sequences. Hence, the model is capable of removing a modest level of noise from previously unseen time series. Unfortunately, the training scores tell a different story. Although the MAE scores are still indicative of the models' denoising abilities, the $R2$ and $RMSE$

ICCS Camera Ready Version 2024
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-63778-0_18



(a) Quantum QAE (Qiskit). (b) Classical AE (Torch).

Fig. 8: Reintegration of time series for quantum and classical models, showing the original pure data (solid line), the noisy input (lighted dotted line), and reconstructed data with reduced noise on output (dashed line).

scores, which are more sensitive to the presence of outliers, suggest that the model had difficulty with noise removal in training. In contrast, the classical models' reconstructions fitted pure data very closely, eliminating virtually all noise in the process. However, their denoising performance on previously unseen data was no better than that of the quantum models.

At the end of this discussion, it is important to take a closer look at the reintegrated time series (see Figure 8). We can observe that the QuTSAE reconstructions (dashed line) seem to follow the shape of the noisy input (dotted line) (see Figure 8a). However, the classical PyTorch reconstructions (dashed line) follow the shape of the pure data (solid line) (see Figure 8b). However, the validation performance of both models is similar. The likely explanation for this phenomenon is that the QuTSAE cost function did not allow the model to fully learn, which can be confirmed by investigating the slope of the cost vs. performance scores over time (compare Figures 6 and 7a). This may have been caused by the lack of nonlinear activations within the quantum model structure [22]. PyTorch models, in turn, converged very quickly; however, the algorithm had insufficient training data to avoid overfitting. Neither of the two models had the opportunity to reach its full potential.

4 Conclusions

Quantum time series analysis is at the intersection of classical time series analysis and quantum machine learning. This article discussed a specific area of quantum time series processing, concerning the application of quantum autoencoders to the elimination of noise from temporal data. The article explained how different design decisions impact the function and performance of quantum time series autoencoders (QuTSAE). In this final section, we provide some reflections on the key design choices for quantum models in general, which could guide and assist researchers and developers of quantum machine learning models.

Requirements. In the initial steps of a quantum model development, one must consider not only the model structure and algorithms suitable for its processing (e.g. the need for the encoder and decoder), but also requirements (e.g. ability of

ICCS Camera Ready Version 2024
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-63778-0_18

Table 2: Final QAE model performance (lat=7, aw=3, reps=2, ep=2000)

Training Accuracy (no edge distortion)			
R2 (org-pure, in-noisy) =	0.782	R2 (org-pure, out-rec) =	0.767
RMSE (org-pure, in-noisy) =	0.078	RMSE (org-pure, out-rec) =	0.081
MAE (org-pure, in-noisy) =	0.064	MAE (org-pure, out-rec) =	0.059
MAPE (org-pure, in-noisy) =	3.985	MAPE (org-pure, out-rec) =	3.592
Validation Accuracy (no edge distortion)			
R2 (org-pure, in-noisy) =	0.795	R2 (org-pure, out-rec) =	0.803
RMSE (org-pure, in-noisy) =	0.076	RMSE (org-pure, out-rec) =	0.074
MAE (org-pure, in-noisy) =	0.064	MAE (org-pure, out-rec) =	0.058
MAPE (org-pure, in-noisy) =	1.962	MAPE (org-pure, out-rec) =	1.421

handling temporal data) and constraints (e.g. level of noise in data not exceeding 5%) imposed on the structure and function of any acceptable solution.

Input. Input encoding schemes (such as QuTSAE *angle encoding around* $|+\rangle$) must match the strategy to measure and interpret the model output (e.g. interpretation of measurements into qubit angular states). The adopted design choices may restrict the use of certain devices for model execution, e.g. QPU or GPU.

Output. Quality design of quantum model observables, their measurement, and interpretation of results are essential for training and testing the model. Design of a suitable cost function (such as swap test or zero testing) is also of pivotal importance, and its use by an optimizer will determine the model trainability, and ultimately its capabilities (such as denoising), size (e.g. width and depth), and efficiency (due to complexity of obtaining and interpreting results) of its architectural options (e.g. half-QAE or full-QAE in QuTSAE).

Ansatz Design. Quantum toolkits offer rich libraries of ansatze suitable for the design of model components (such as the QuTSAE encoder and decoder). It is also possible to hand-craft a parameterized custom ansatz. Several important objectives in ansatz design must be considered. The ansatz structural properties must fit the model function (such as the need for QuTSAE symmetry). The rotational and entanglement blocks must match the input encoding and output decoding methods (the simplicity and compatibility of the *RealAmplitude* ansatz vs. the versatility of the *TwoLocal* ansatze in QuTSAE). The ansatz design, with its width, depth, the number of trainable parameters, as well as the required degree of freedom, will determine the suitability of its optimization strategies.

Model Training and Validation. Virtually all design choices for quantum models will have a major impact on their performance in training and validation. Typically, a series of experiments should be designed to establish which design aspects could influence model performance. To this end, an overarching experimental framework may need to be introduced, with data repositories and development tools (e.g. IRI marketing data, Qiskit and PyTorch toolkits), measuring instruments for model scoring and analysis (e.g. R^2, $RMSE$, MAE and $MAPE$), results capture and presentation facilities (e.g. MLFlow), and eventually source code and results distribution (e.g. via github).

Objectives and Success Criteria. From the outset, it is necessary to clearly define quantum project objectives (e.g. analysis of temporal data) and a set of

ICCS Camera Ready Version 2024
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-63778-0_18

testable success criteria (e.g. the target level of noise reduction in data). It is important to clearly state what goals are aspirational only and hence out of scope (such as the quantum model's ability to detect and remove "complex noise patterns"), but which could be investigated in the future.

As is evident from this article, quantum model development must constantly balance the quality and precision of the results with the complexity of developed models and processes, the available resources, and the practical execution time to produce useful results. It is a difficult process to manage and master. Nevertheless, the research and development process that we followed in this project provided us with some valuable learning experience.

We believe that our insights, as reported in this article, are transferable to other researchers and developers, other types of quantum time series analysis model, and quantum models in general, using different data, with distinct objectives, and a variety of quantum and machine learning tools.

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Bajaj, A.: Anomaly Detection in Time Series. https://neptune.ai/blog/anomaly-detection-in-time-series (Mar 2021)
2. Bausch, J.: Recurrent quantum neural networks. Advances in neural information processing systems **33**, 1368–1379 (2020)
3. Berahmand, K., Daneshfar, F., Salehi, E.S., Li, Y., Xu, Y.: Autoencoders and their applications in machine learning: A survey. Artificial Intelligence Review **57**(2), 28 (Feb 2024)
4. Blázquez-García, A., Conde, A., Mori, U., Lozano, J.A.: A Review on Outlier/Anomaly Detection in Time Series Data. ACM Computing Surveys (CSUR) **54**(3), 1–33 (2021)
5. Bravo-Prieto, C.: Quantum autoencoders with enhanced data encoding. Machine Learning: Science and Technology **2** (May 2021)
6. Bronnenberg, B.J., Kruger, M.W., Mela, C.F.: Database Paper —The IRI Marketing Data Set. Marketing Science **27**(4), 745–748 (Jul 2008)
7. Cerezo, M., Arrasmith, A., Babbush, R., Benjamin, S.C., Endo, S., Fujii, K., McClean, J.R., Mitarai, K., Yuan, X., Cincio, L., Coles, P.J.: Variational quantum algorithms. Nature Reviews Physics **3**(9), 625–644 (2021)
8. Chen, S.Y.C., Yoo, S., Fang, Y.L.L.: Quantum long short-term memory. In: ICASSP 2022-2022 IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP). pp. 8622–8626. IEEE (2022)
9. Chen, S., Guo, W.: Auto-Encoders in Deep Learning—A Review with New Perspectives. Mathematics **11**(8), 1777 (Jan 2023)
10. Chiarot, G., Silvestri, C.: Time series compression: A survey. ACM Computing Surveys **55**(10), 1–32 (Oct 2023)
11. Eldar, Y., Oppenheim, A.: Quantum signal processing. IEEE Signal Processing Magazine **19**(6), 12–32 (Nov 2002)
12. Emmanoulopoulos, D., Dimoska, S.: Quantum Machine Learning in Finance: Time Series Forecasting. Tech. Rep. arXiv:2202.00599, arXiv (Feb 2022)

ICCS Camera Ready Version 2024
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-63778-0_18

13. Goodfellow, I., Bengio, Y., Courville, A.: Deep Learning. The MIT Press (2016)
14. Hyndman, R.J., Athanasopoulos, G.: Forecasting: Principles and Practice, 3rd Edition. OTexts, Melbourne, Australia (2021)
15. Khoshaman, A., Vinci, W., Denis, B., Andriyash, E., Sadeghi, H., Amin, M.H.: Quantum variational autoencoder. Quantum Science and Technology **4**(1), 014001 (Sep 2018)
16. Li, P., Pei, Y., Li, J.: A comprehensive survey on design and application of autoencoder in deep learning. Applied Soft Computing **138**, 110176 (May 2023)
17. Liu, C.J., Liu, H.T., Bian, C., Chen, X.D., Yang, S.H., Wang, X.F.: Investigation of Time-series Prediction for Turbine Machinery Condition Monitoring. IOP Conference Series: Materials Science and Engineering **1081**(1), 012022 (Feb 2021)
18. Liu, N., Rebentrost, P.: Quantum machine learning for quantum anomaly detection. Physical Review A **97**(4), 042315 (2018)
19. Lu, L., Yin, K.L., de Lamare, R.C., Zheng, Z., Yu, Y., Yang, X., Chen, B.: A survey on active noise control in the past decade—Part I: Linear systems. Signal Processing **183**, 108039 (Jun 2021)
20. Lu, L., Yin, K.L., de Lamare, R.C., Zheng, Z., Yu, Y., Yang, X., Chen, B.: A survey on active noise control in the past decade–Part II: Nonlinear systems. Signal Processing **181**, 107929 (Apr 2021)
21. Markov, V., Rastunkov, V., Fry, D.: Quantum Time Series Similarity Measures and Quantum Temporal Kernels (Dec 2023)
22. Maronese, M., Destri, C., Prati, E.: Quantum activation functions for quantum neural networks. Quantum Information Processing **21**(4), 128 (Apr 2022)
23. Pérez-Salinas, A., Cervera-Lierta, A., Gil-Fuster, E., Latorre, J.I.: Data re-uploading for a universal quantum classifier. Quantum **4**, 226 (Feb 2020)
24. Rivas, P., Zhao, L., Orduz, J.: Hybrid Quantum Variational Autoencoders for Representation Learning. In: 2021 Int. Conf. on Computational Science and Computational Intelligence (CSCI). pp. 52–57. IEEE, Las Vegas, NV, USA (Dec 2021)
25. Romero, J., Olson, J.P., Aspuru-Guzik, A.: Quantum autoencoders for efficient compression of quantum data. Quantum Science and Technology **2**(4), 045001 (2017)
26. Sakhnenko, A., O'Meara, C., Ghosh, K.J., Mendl, C.B., Cortiana, G., Bernabé-Moreno, J.: Hybrid classical-quantum autoencoder for anomaly detection. Quantum Machine Intelligence **4**(2), 27 (2022)
27. Schuld, M., Petruccione, F.: Machine Learning with Quantum Computers. Springer, 2nd ed. 2021 edition edn. (Oct 2021)
28. Sezer, O.B., Gudelek, M.U., Ozbayoglu, A.M.: Financial Time Series Forecasting with Deep Learning : A Systematic Literature Review: 2005-2019 (Nov 2019)
29. Takaki, Y., Mitarai, K., Negoro, M., Fujii, K., Kitagawa, M.: Learning temporal data with variational quantum recurrent neural network. Physical Review A **103**(5), 052414 (May 2021)
30. Tripathi, P.M., Kumar, A., Komaragiri, R., Kumar, M.: A Review on Computational Methods for Denoising and Detecting ECG Signals to Detect Cardiovascular Diseases. Archives of Computational Methods in Engineering **29**(3), 1875–1914 (May 2022)
31. Weigold, M., Barzen, J., Leymann, F., Salm, M.: Encoding patterns for quantum algorithms. IET Quantum Communication **2**(4), 141–152 (2021)
32. Yarkoni, S., Kleshchonok, A., Dzerin, Y., Neukart, F., Hilbert, M.: Semi-supervised time series classification method for quantum computing. Quantum Machine Intelligence **3**(1), 12 (Apr 2021)

ICCS Camera Ready Version 2024
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-63778-0_18