The Art of Data Encoding and Decoding for Quantum Time Series Analysis

Jacob L. Cybulski
Deakin University
Melbourne, Australia
jacob.cybulski@deakin.edu.au

Sebastian Zając
SGH Warsaw School of Economics
Warsaw, Poland
szajac2@sgh.waw.pl

Abstract: *Time series (TS) feature unstructured data and unique processing, thus, they do not easily fit conventional approaches to quantum modelling.*

This project thus investigated various approaches to encoding and (in part) decoding of time series data to support quantum time-series analysis (QTSA), which relied on curve-fitting and forecasting with sliding-windows (SW).

As each approach has the potential to enhance or impede the model's ability to represent and manipulate data, this project therefore compared the selected approaches and evaluated their performance using a noise-free quantum simulator. It did not seek quantum advantage of these methods.

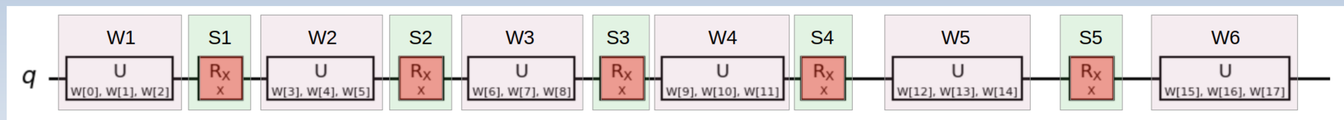
*The best models used similar methods of training and the same data decoding: with all n qubits measured and using ("Z"*n, 1.0) observables.*

Platform: Ubuntu 22.04, Python 3.11, Qiskit 1.2.4, QML 0.7.2, Aer-gpu 0.15.1
All models: EstimatorQNN + NNRegressor, 1-2 observables, Optimiser: L_BBFGS_B + L2Loss

Considered data encoding approaches

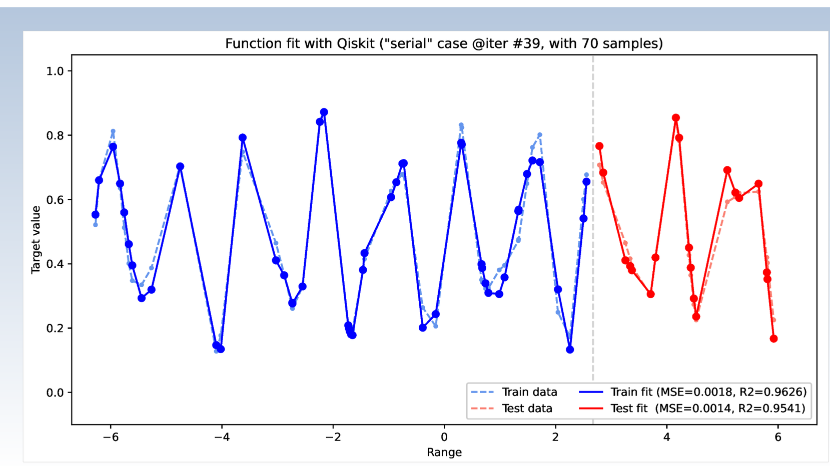
Synthetic "oscillation" data with 70 samples:
f(x) = (sin(5x) + 0.5*sin(8x)) / 4 + 0.5

PQFT serial reuploading model: This is a curve fitting model mapping x to the value of f(x). It implements Partial Quantum Fourier Transform on a single qubit. The model parameters are state rotations (Rx, Ry, Rz). Input data, angle encoded with Rx, is being serially reuploaded between parameter blocks.



Sample PQFT reuploading serial model
Actual model: 1 qubit, 27 layers, 84 weights

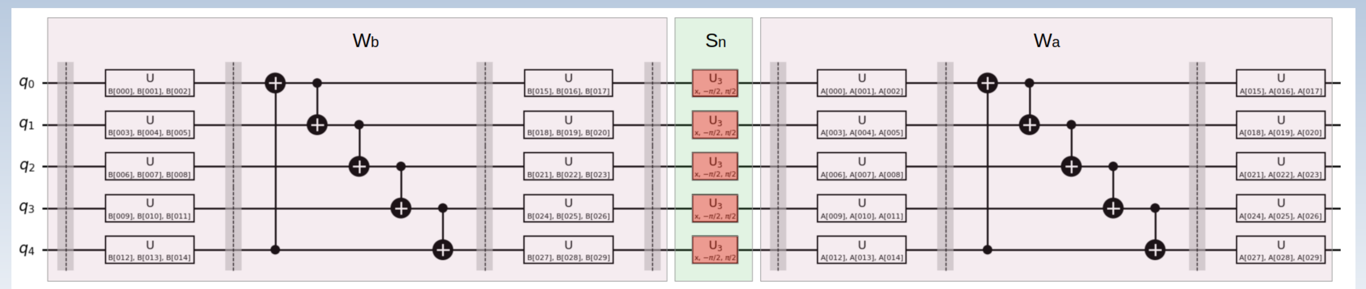
Oscillation data favors PQFT models!



All models were tested in a variety of configurations

PQFT serial models performed closest to the level of classical NNs. They fitted "oscillation" data with very high level of accuracy.

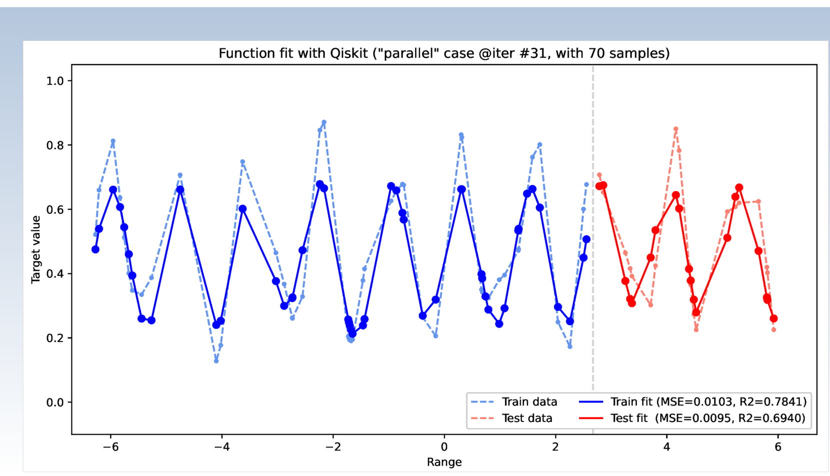
PQFT parallel reuploading model: This is a curve fitting model mapping x to the value of f(x). It implements PQFT on multiple entangled qubits. It is often claimed to be an improvement over PQFT serial models. Its data block reuploads input to each qubit in parallel, and is surrounded by multiple layers of (Rx, Ry, Rz) parameters and entangling blocks.



circular entanglement

slow training

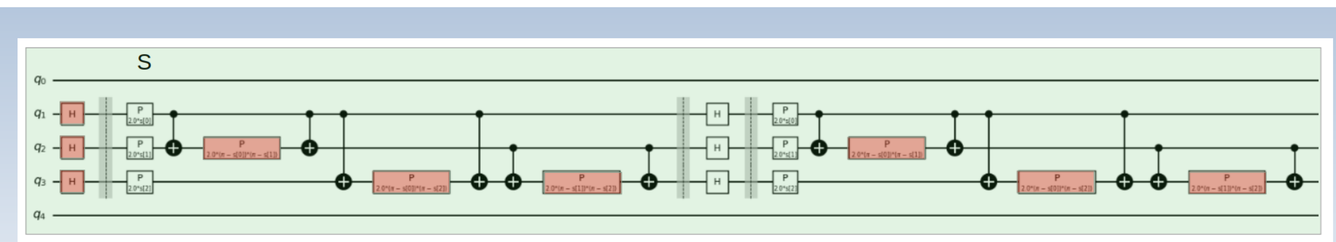
Sample PQFT reuploading parallel model
Actual model: 5 qubits, 3+3 layers, 120 weights



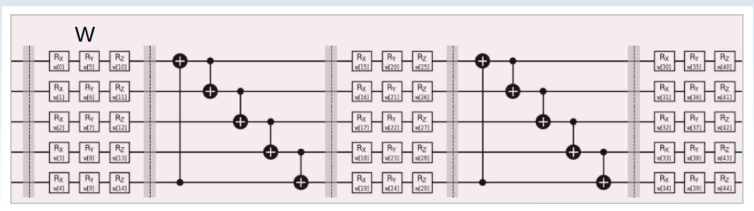
PQFT parallel models failed to improve on the performance of their serial counter-parts. Also, their TS fit could not excel the fit of other TS models. Due to the large number of parameters needed, their training was very very slow.

SW QNN standard model: A QNN forecasting model, based on a sliding-window of size 5, and horizon of 1. It has a feature map and an ansatz of (Rx, Ry, Rz) state rotations. An ansatz can add extra width to the circuit. When using an amplitude encoding, the number of qubits drops to log2 of data features. In angle- or phase-encoding (as in ZZ feature maps), the model requires as many qubits as the window length.

ZZ fmap

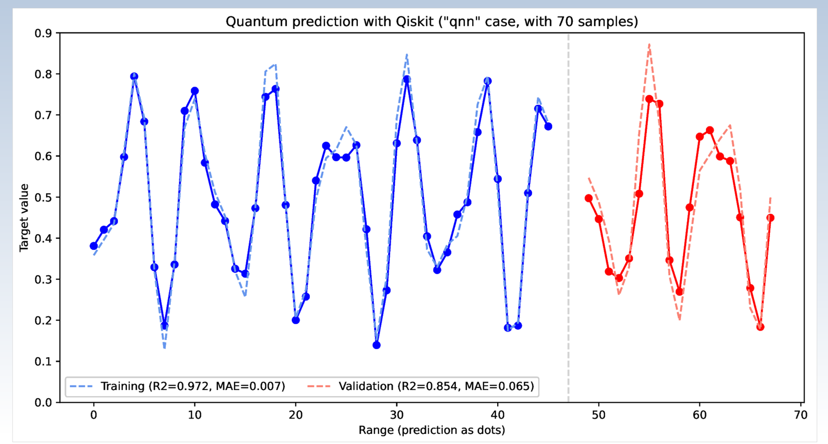


slow training



circular entanglement

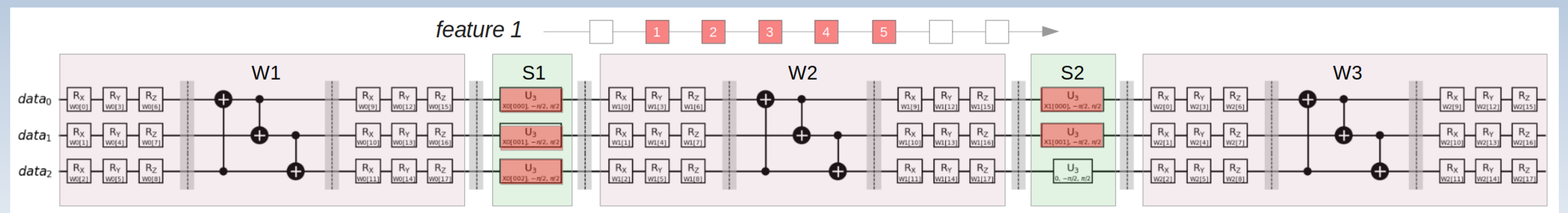
Sample SW QNN model (shows the ansatz wider than the fm)
Actual model: 9 qubits, 5 inputs, 1 fm + 5 ansatz layers, 162 weights



This chart shows average of overlapping windows

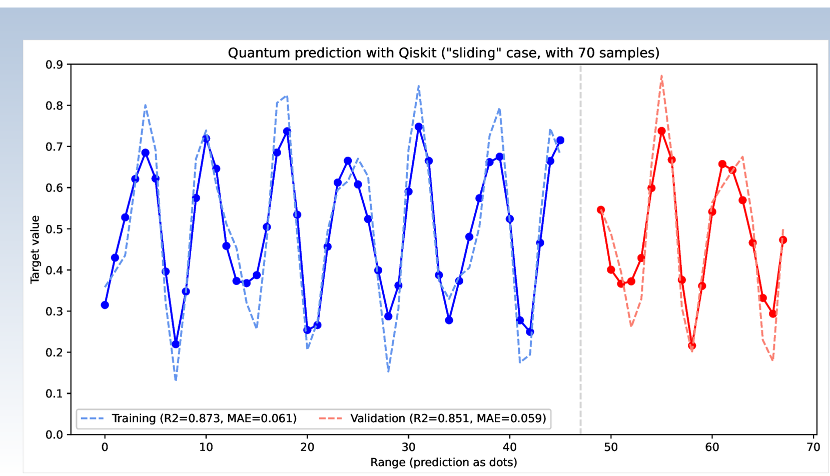
When adding weights, SW QNNs training and the ability to generalise improved. As amplitude encoding varies the model structure with data, model training with GPUs/QPUs was not possible. ZZ-fms were used instead, esp. that the circuit needed extra weights by increasing its width.

SW QTSA overloading model: A forecasting model, with sliding-window of size 5, and horizon of 1. The model consists of multiple layers, each reuploading a TS window into a series of encoding blocks. The blocks overload qubits with different window parts. The approach fuses PQFT ideas with the SW QNN model. This model can also encode multi-variate TSs.



circular entanglement

Sample QTSA uni-variate overloading model (shows 1 layer of 2 blocks)
Actual model: 3 qubits, 2 layers of 2 encoding blocks each, 90 weights



This chart shows average of overlapping windows

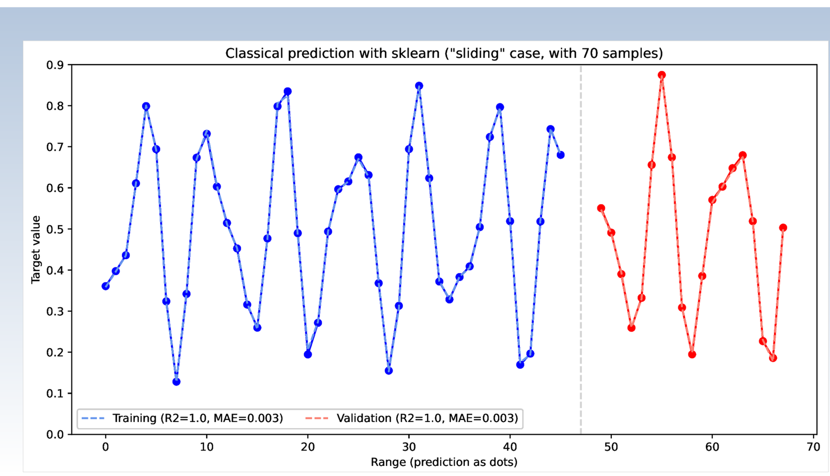
The SW QTSA performance was second best in training and testing. Many experiments were needed to identify the best model, especially in the design of quantum observables. Due to qubit overloading, the number of qubits was reduced.

Conclusions: *This project explored the impact of data encoding strategy on QTSA models. Heavily parameterised models that used data reuploading performed well in scoring and qubit utilisation. To our surprise, 1 qubit PQFT serial models were the best performers. SW QTSA and SW QNNs performed equally well in testing, SW QTSA were more qubit efficient. SW QNNs benefitted from extra circuit width.*

Model	Qubit#	Param#	Obs	Optimizer	Epochs	Training R2	Training MAE	Testing R2	Testing MAE
SW MLP Sklearn	-	21101	-	LBFGS	850	0.999	0.003	0.999	0.003
Serial PQFT	1	84	1	L-BFGS-B	39	0.971	0.029	0.954	0.029
Parallel PQFT	5	120	1	L-BFGS-B	31	0.862	0.071	0.694	0.081
SW QTSA	3	90	1	L-BFGS-B	65	0.873	0.060	0.851	0.058
SW QNN	9	162	1	L-BFGS-B	32	0.998	0.007	0.854	0.060

Sorted by test scores.
Epochs @ best test MAE.
In SW models, scores compare expected vs predicted window values, with overlapping window averaged by TS position.

PQFT and SW models used the same data, but of different structures and size.
They can only be compared with R2 scores, within their groups also MAE.
By test R2, serial PQFTs performed best, SW QTSA and QNNs were on par.



This chart shows average of overlapping windows

A classical NN model was also developed. The model had a huge number of parameters, however its training was very quick. The model performance was the best and was therefore used to benchmark the quantum models.

QTML'2024