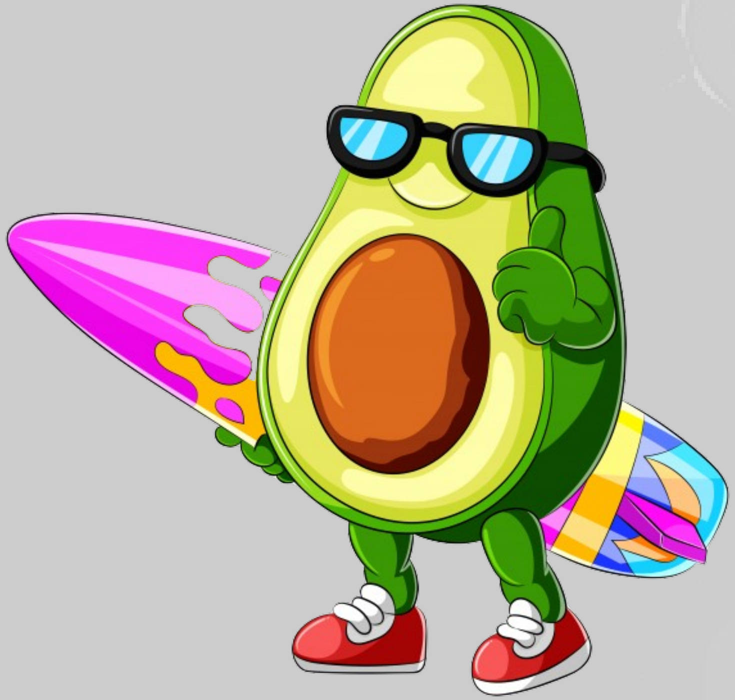


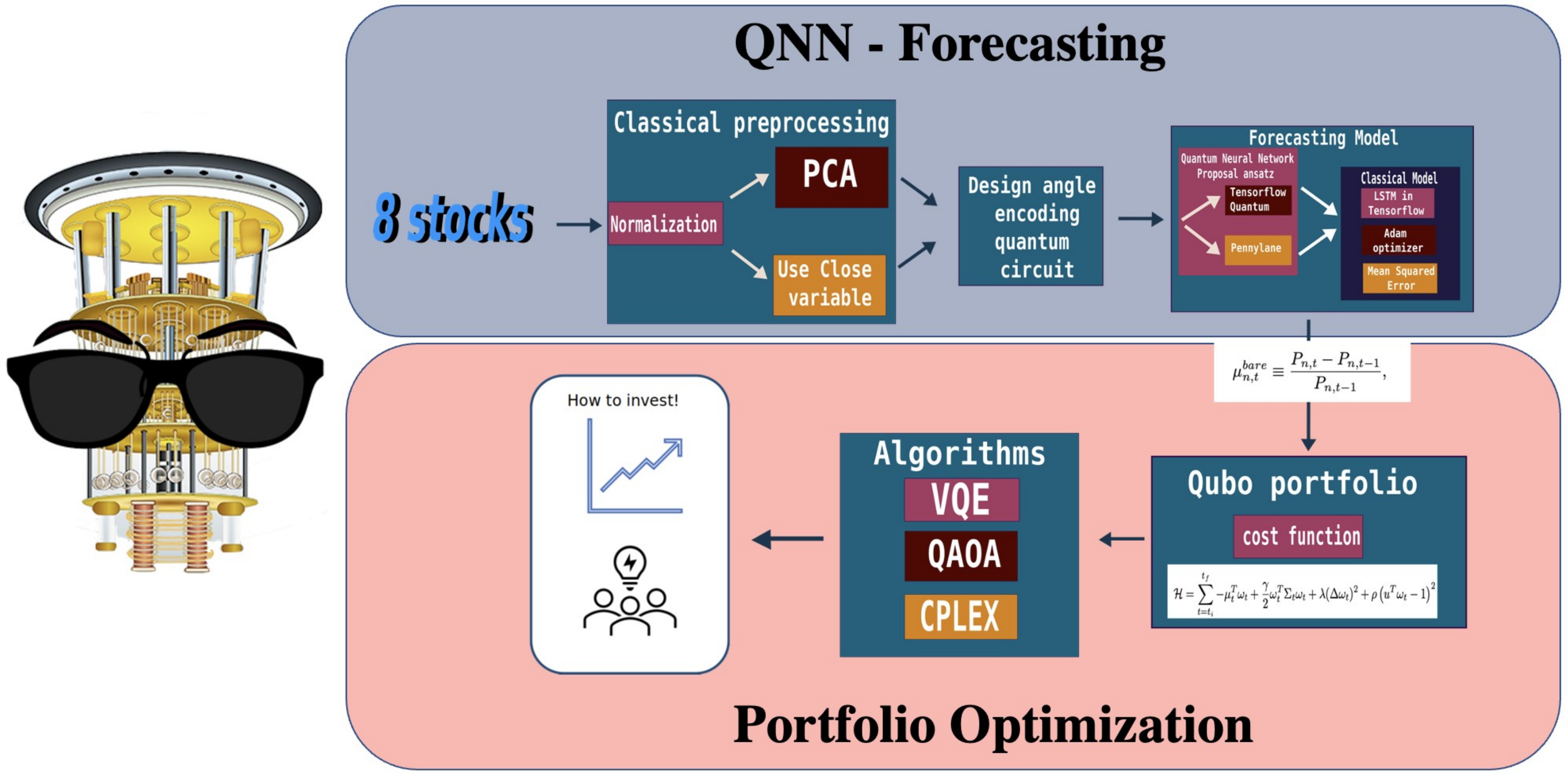
Quantum Counselor for Portfolio investment

Team Avocados

- Alejandro Montañez
- Alberto Maldonado

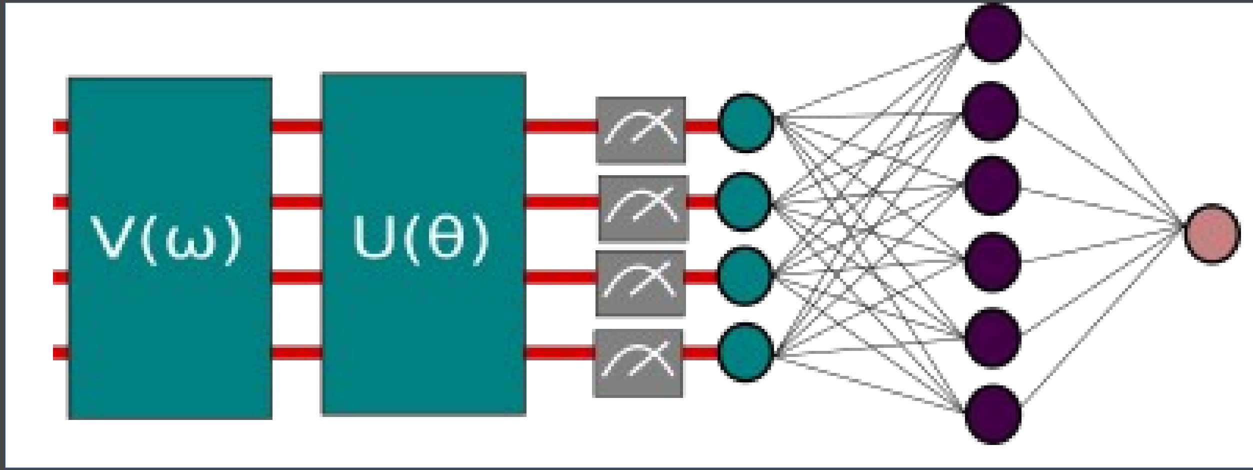
Outline

- 1. Quantum neural network (QNN) for stock forecasting.
- 2. Portfolio optimization
- 3. A novel approach for the Portfolio Optimization
- 4. Conclusion and future work



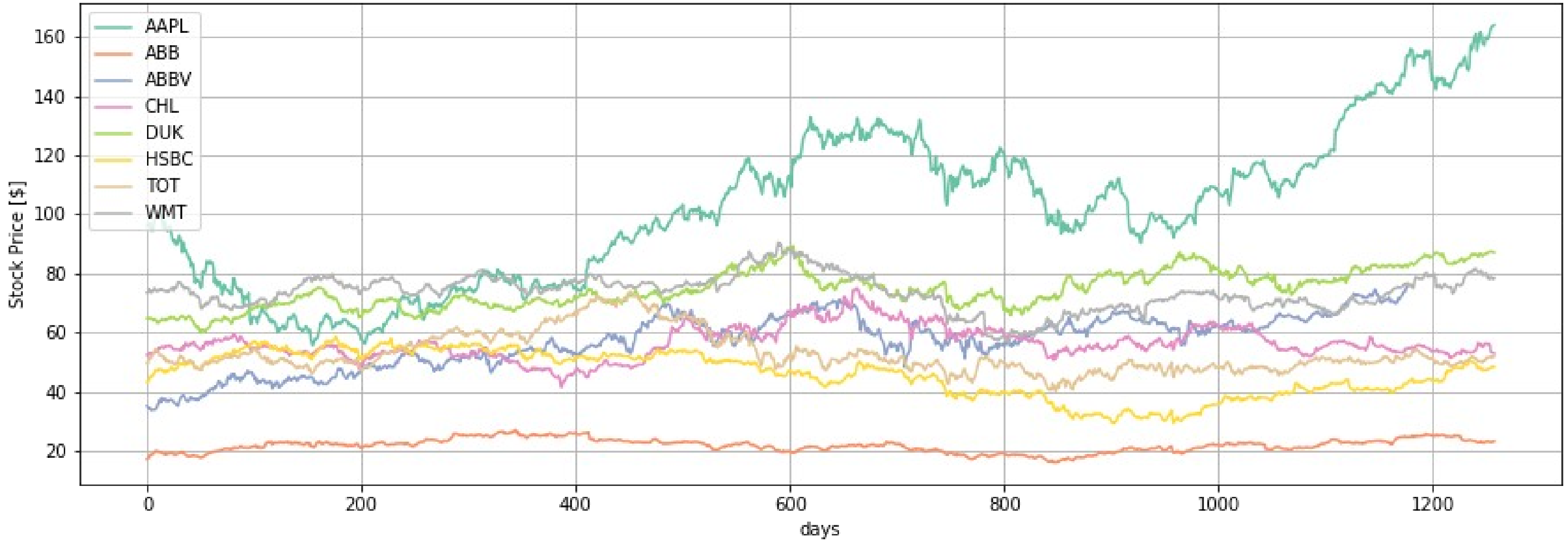
QNN - Forecasting
8 stocks
Classical preprocessing
Normalization
PCA
Use Close variable
Design angle encoding quantum circuit
Forecasting Model
Quantum Neural Network Proposal ansatz
Tensorflow Quantum
Pennylane
Classical Model
LSTM in Tensorflow
Adam optimizer
Mean Squared Error
$\mu_{n,t}^{bare} \equiv \frac{P_{n,t} - P_{n,t-1}}{P_{n,t-1}},$
Qubo portfolio
cost function
$\mathcal{H} = \sum_{t=t_i}^{t_f} -\mu_t^T \omega_t + \frac{\gamma}{2}\omega_t^T \Sigma_t \omega_t + \lambda(\Delta\omega_t)^2 + \rho\left(u^T\omega_t - 1\right)^2$
Algorithms
VQE
QAOA
CPLEX
How to invest!
Portfolio Optimization

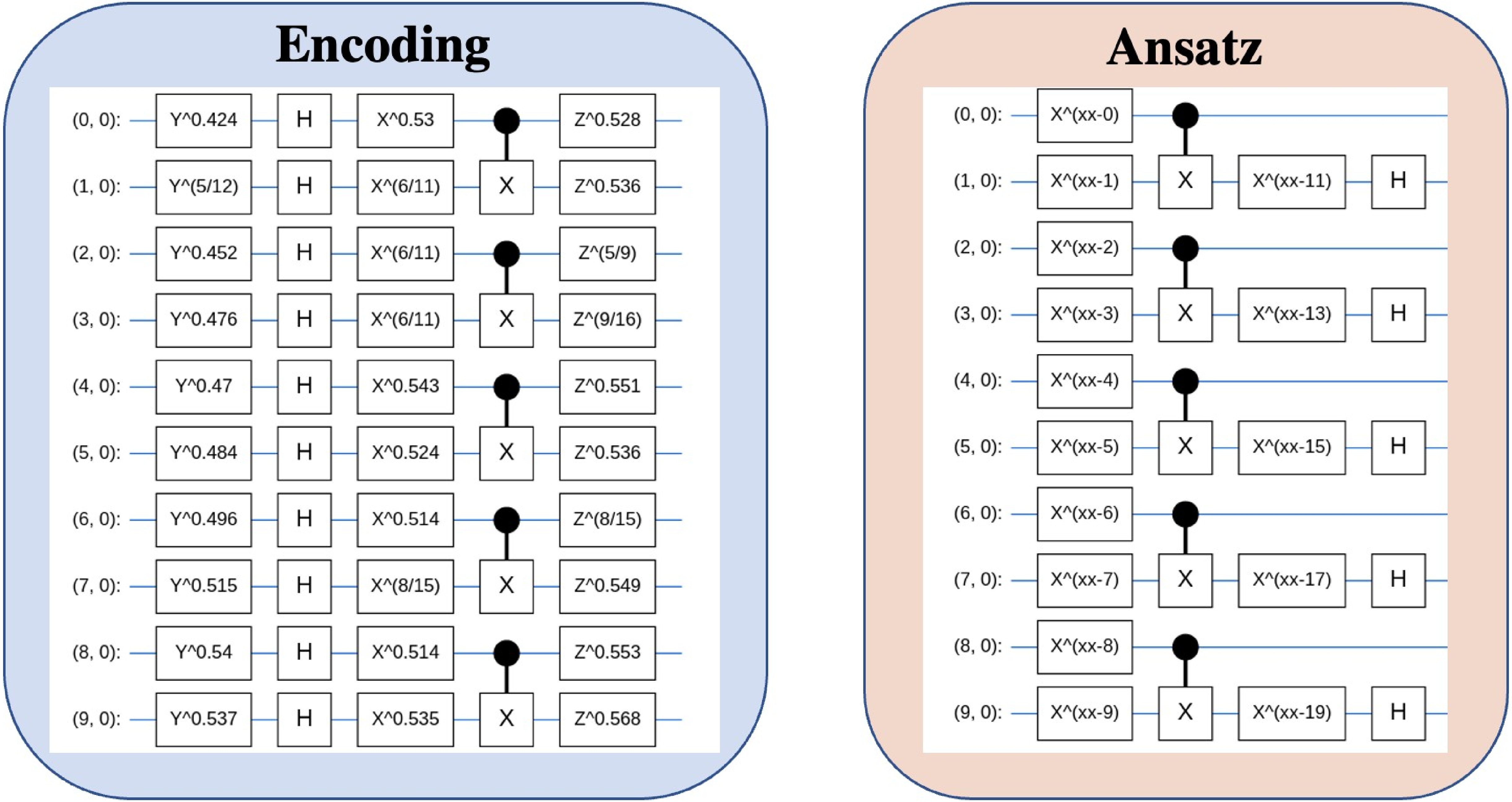
1. Stocks forecasting using a QNN



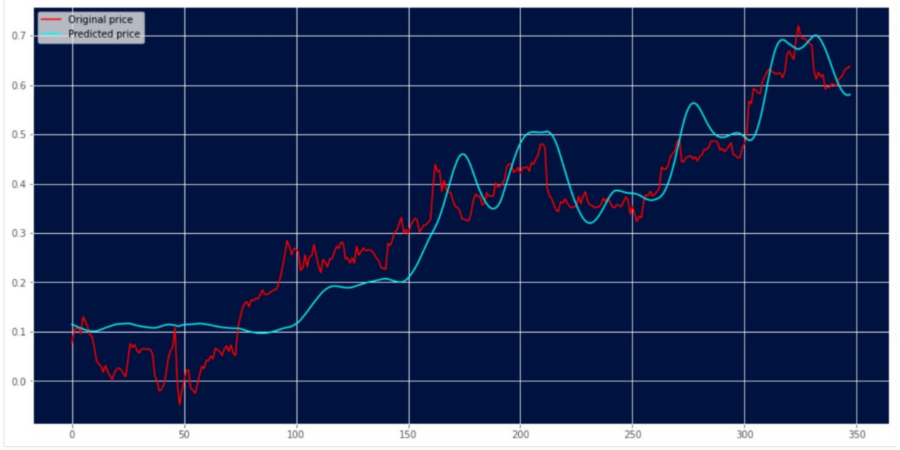
Stocks

Basic Materials: TOTAL S.A. "TOT"
Consumer Goods: Appel Inc. "AAPL"
Healthcare: AbbVie Inc. "ABBV"
Services: Wall-Mart Stores Inc. "WMT"
Utilites: Duke energy corporation "DUK"
Financial: HSBS Holding pcl "HSBC"
Industrial Goods: ABB Ltd. "ABB"
Technology: China Mobile Limited "CHL"

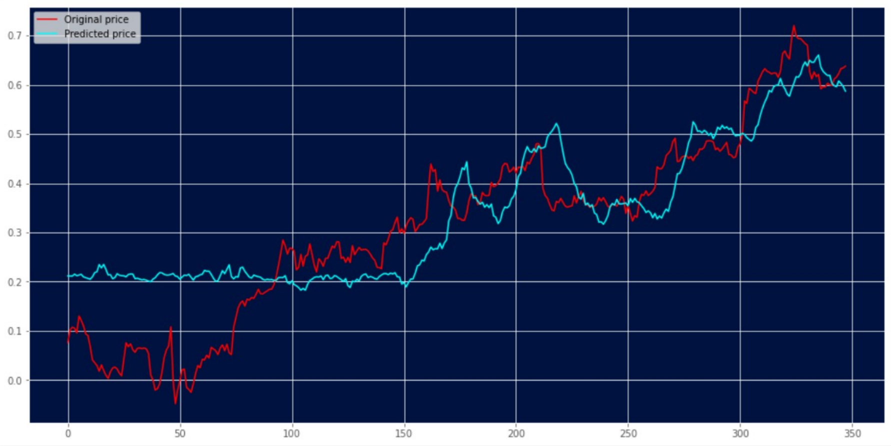



Encoding
(0, 0): Y^0.424 H X^0.53 Z^0.528
(1, 0): Y^(5/12) H X^(6/11) X Z^0.536
(2, 0): Y^0.452 H X^(6/11) Z^(5/9)
(3, 0): Y^0.476 H X^(6/11) X Z^(9/16)
(4, 0): Y^0.47 H X^0.543 Z^0.551
(5, 0): Y^0.484 H X^0.524 X Z^0.536
(6, 0): Y^0.496 H X^0.514 Z^(8/15)
(7, 0): Y^0.515 H X^(8/15) X Z^0.549
(8, 0): Y^0.54 H X^0.514 Z^0.553
(9, 0): Y^0.537 H X^0.535 X Z^0.568
Ansatz
(0, 0): X^(xx-0)
(1, 0): X^(xx-1) X X^(xx-11) H
(2, 0): X^(xx-2)
(3, 0): X^(xx-3) X X^(xx-13) H
(4, 0): X^(xx-4)
(5, 0): X^(xx-5) X X^(xx-15) H
(6, 0): X^(xx-6)
(7, 0): X^(xx-7) X X^(xx-17) H
(8, 0): X^(xx-8)
(9, 0): X^(xx-9) X X^(xx-19) H

Classical Minimal Model

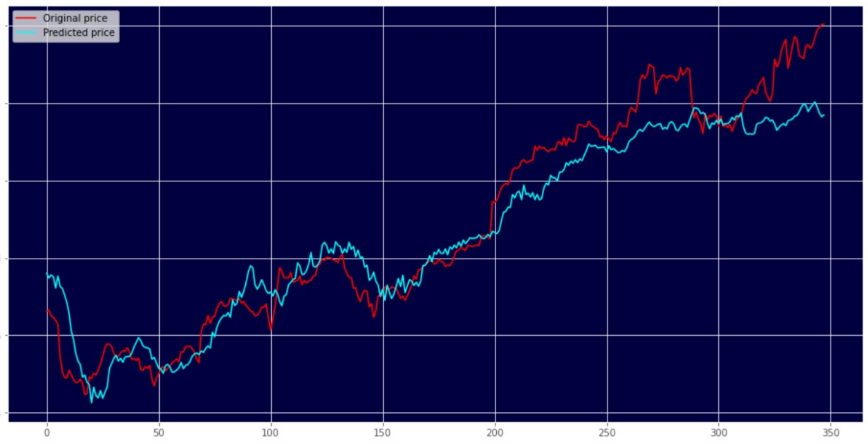


Hybrid Minimal Model

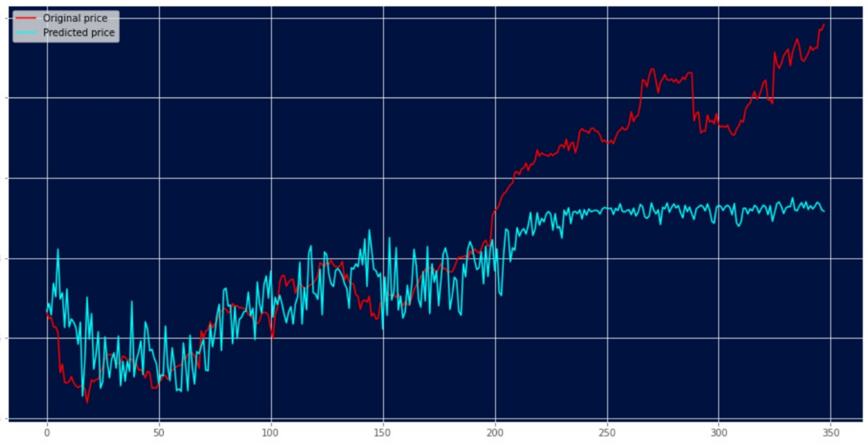


Classical Model

Hybrid Model

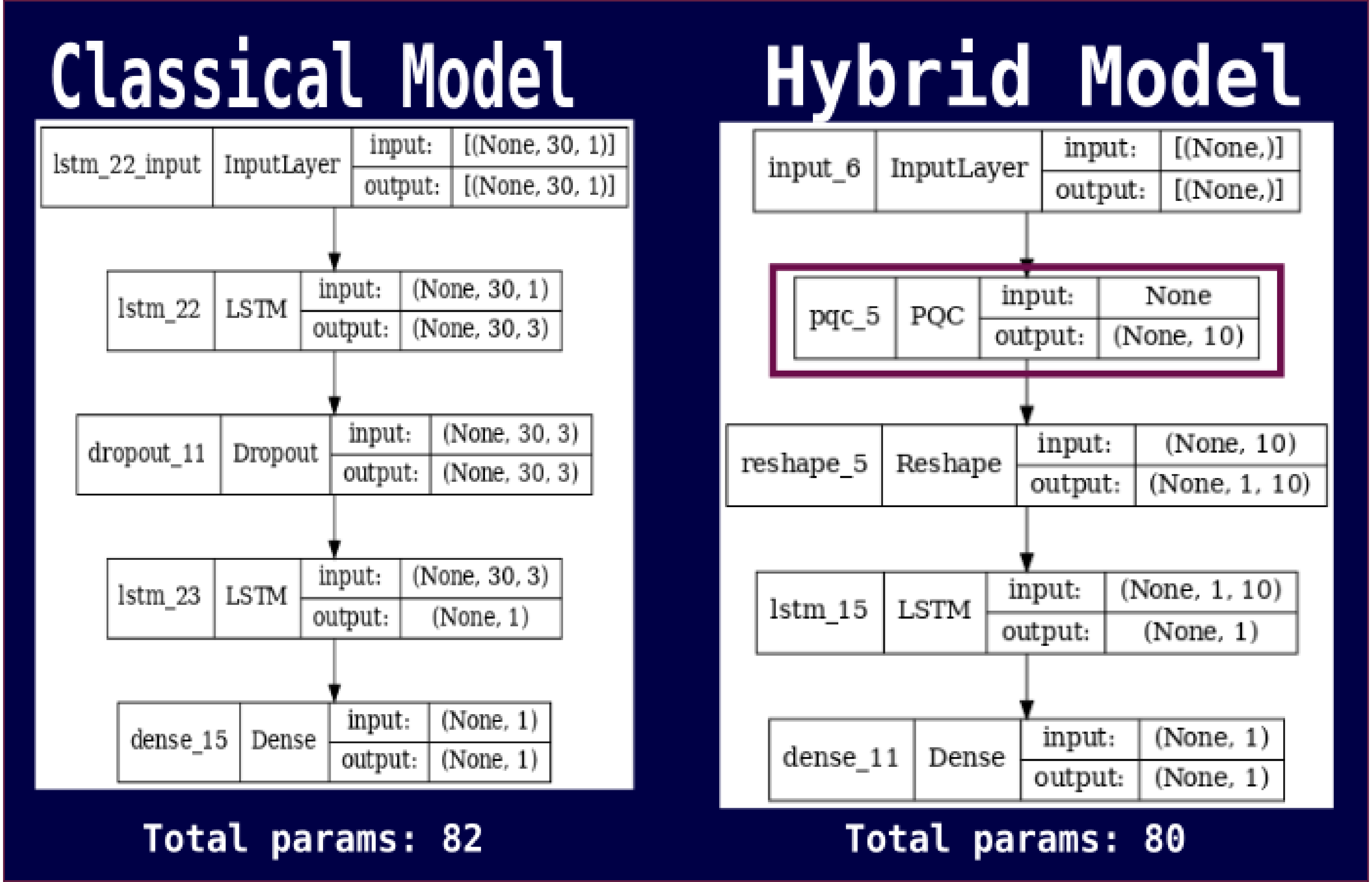


Hybrid Model with Nosie



Red: original

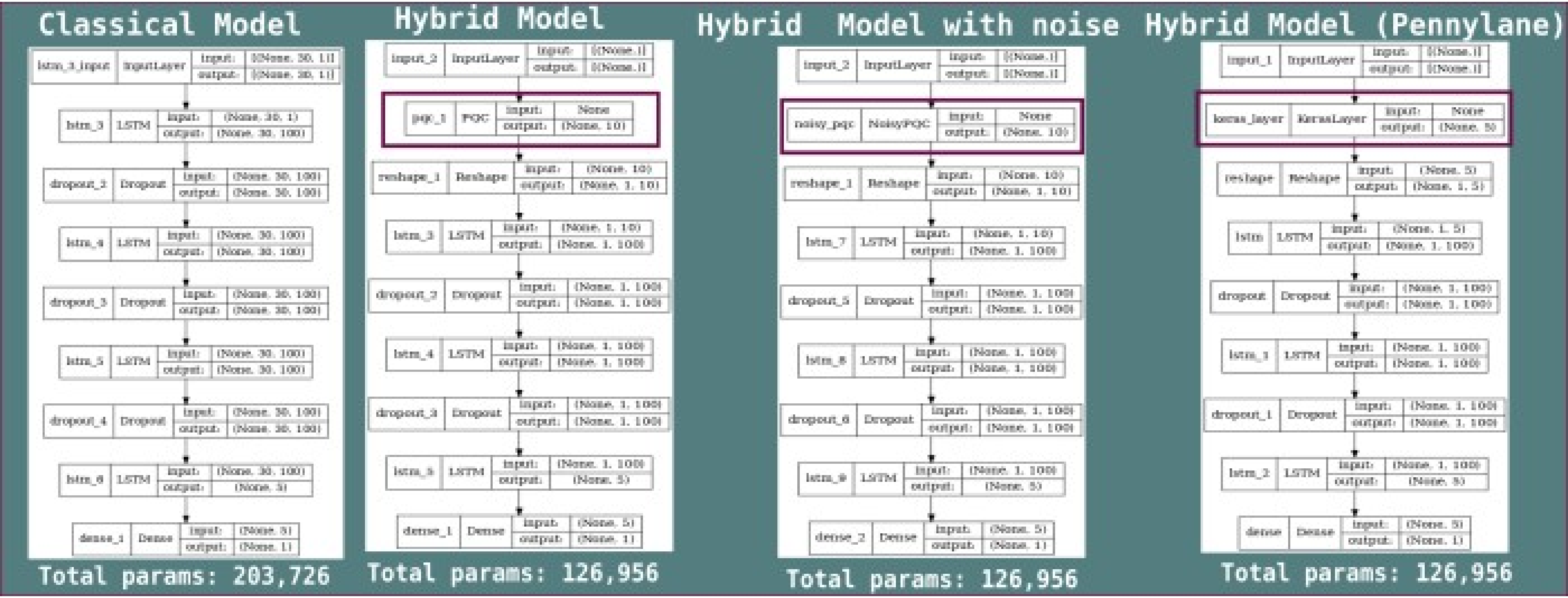
Cyan: prediction



Classical Model
lstm_22_input
InputLayer
input:
[(None, 30, 1)]
output:
[(None, 30, 1)]
lstm_22
LSTM
input:
(None, 30, 1)
output:
(None, 30, 3)
dropout_11
Dropout
input:
(None, 30, 3)
output:
(None, 30, 3)
lstm_23
LSTM
input:
(None, 30, 3)
output:
(None, 1)
dense_15
Dense
input:
(None, 1)
output:
(None, 1)
Total params: 82
Hybrid Model
input_6
InputLayer
input:
[(None,)]
output:
[(None,)]
pqc_5
PQC
input:
None
output:
(None, 10)
reshape_5
Reshape
input:
(None, 10)
output:
(None, 1, 10)
lstm_15
LSTM
input:
(None, 1, 10)
output:
(None, 1)
dense_11
Dense
input:
(None, 1)
output:
(None, 1)
Total params: 80

Mean Absolute Error (MAE) with 1 results using classical and hybrid minimal model

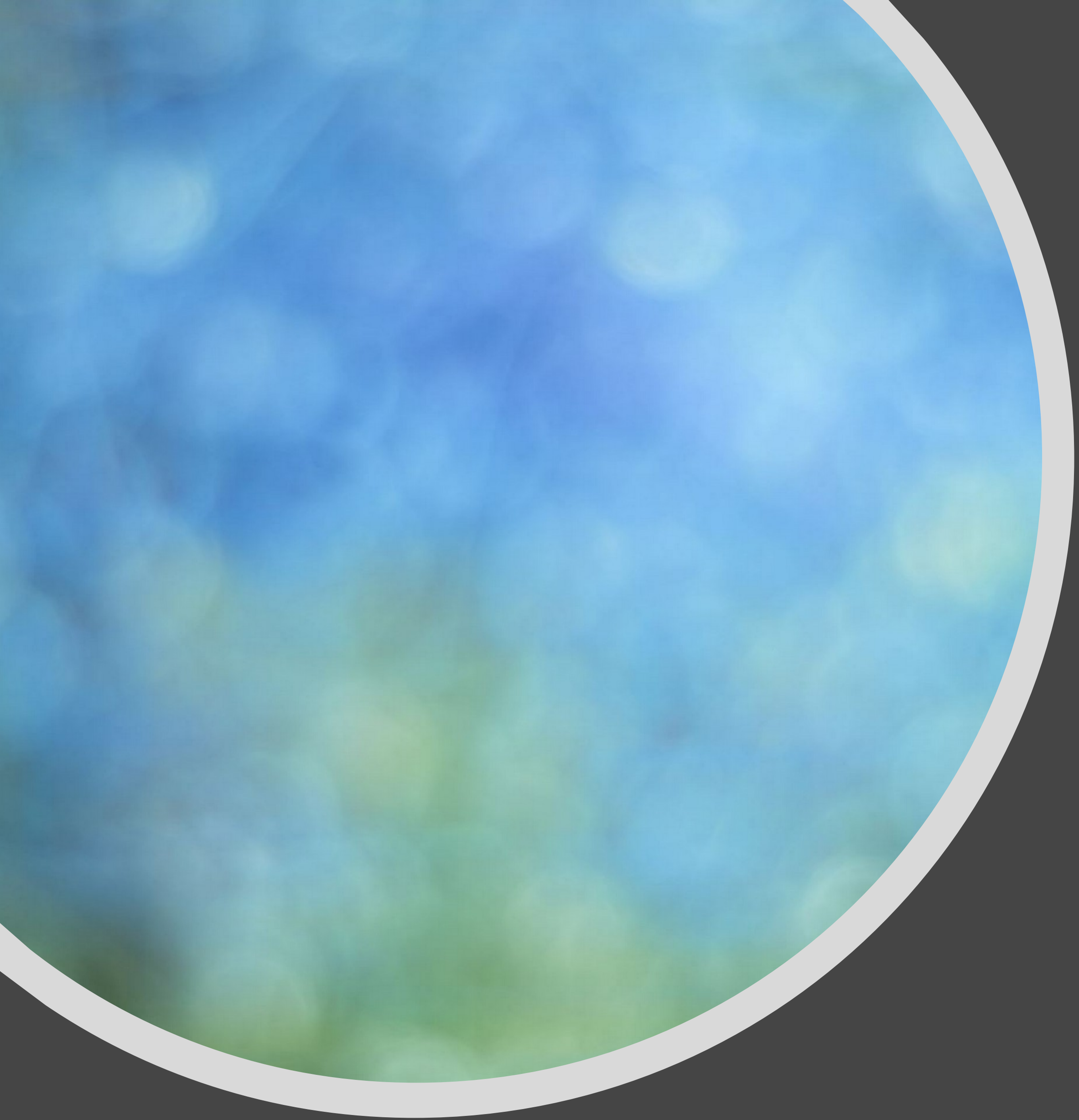
	Not Using PCA				Using PCA			
	Classical		Quantum		Classical		Quantum	
names	Error train (%)	Error test (%)	Error train (%)	Error test (%)	Error train (%)	Error test (%)	Error train (%)	Error test (%)
AAPL	4.341642	4.746031	3.498561	3.569917	3.224976	3.392665	4.225214	4.387063
ABB	4.034601	3.751406	3.124790	3.102234	2.770057	2.741467	3.838684	4.307379
ABBV	4.251284	4.654431	3.923027	4.862839	5.147607	6.593405	3.225815	3.004062
TOT	3.487782	3.147850	3.646976	3.671802	4.137933	3.661434	3.547327	3.271123
WMT	3.839625	3.644914	3.577528	3.393435	4.391325	4.391842	5.435431	5.294369
DUK	3.856109	4.195548	4.095765	3.984043	7.286841	5.018786	4.140656	3.777807
CHL	3.921305	3.422907	5.811381	3.508788	5.724068	3.098087	7.014563	5.308409
HSBC	4.098955	5.923856	4.400225	7.820905	3.645569	5.850563	3.866826	4.138372


Classical Model
Hybrid Model
Hybrid Model with noise
Hybrid Model (Pennylane)
Total params: 203,726
Total params: 126,956
Total params: 126,956
Total params: 126,956

Mean Absolute Error (MAE) with 10 results using classical and hybrid model

	Not Using PCA				Using PCA			
	Classical		Quantum		Classical		Quantum	
names	Error train (%)	Error test (%)	Error train (%)	Error test (%)	Error train (%)	Error test (%)	Error train (%)	Error test (%)
AAPL	2.413642	6.356331	2.350965	7.581864	2.489594	6.074156	2.581829	7.997429
ABB	2.570945	2.396095	2.774117	2.475377	2.624829	2.552587	2.877535	2.762380
ABBV	2.784104	2.686317	2.670232	2.967503	2.719091	2.681797	2.735113	2.851575
TOT	2.436926	1.968772	2.509338	1.862748	2.557738	2.206377	2.698185	1.996073
WMT	2.317517	2.284026	2.501853	2.575917	2.452581	2.324351	2.682050	2.716340
DUK	2.456103	2.477246	2.911269	3.188830	2.428863	2.594698	2.725936	3.291306
CHL	2.430171	1.874603	2.652121	2.039861	2.274212	1.720072	2.514547	1.930191
HSBC	2.802238	3.035687	2.691497	3.714446	2.698808	3.111532	2.609961	3.295598

Time per epoch for each of the 4 models (sec)			
Classical Model	Hybrid Model	Hybrid Model with noise	Hybrid Model (Pennylane)
1 sec	3 sec	33 sec	121 sec



2. Portfolio optimization

- Model XS (3 Stocks, 2 periods), QAOA and VQE with SPSA and COBYLA classical optimizers.

- Model S (5 Stocks, 3 periods), QAOA and VQE with SPSA and COBYLA classical optimizers.

- Model M (8 Stocks, 3 periods), QAOA and VQE with SPSA and COBYLA classical optimizers.

Cost Function

$$\mathcal{H} = \boxed{\sum_{t=t_i}^{t_f} -\mu_t^T \omega_t} + \boxed{\frac{\gamma}{2}\omega_t^T \Sigma_t \omega_t} + \boxed{\lambda(\Delta\omega_t)^2} + \boxed{\rho(u^T\omega_t - 1)^2}$$

Return **Risk** **Transaction Cost** **Budget constraint**

$$\omega_t = \begin{bmatrix} \omega_{0,t} \\ \vdots \\ \omega_{N,t} \end{bmatrix}$$

is a vector with the percentage of investment for N assets at time t.

$$\mu_t = \begin{bmatrix} \mu_{0,t} \\ \vdots \\ \mu_{N,t} \end{bmatrix} \rightarrow \mu_{n,t} = \sum_{n=1}^{N} \frac{P_{n,t} - P_{n,t-1}}{P_{n,t-1}}$$

is the bare return for each asset and is the price at time t of asset n.

$$\Sigma_t$$

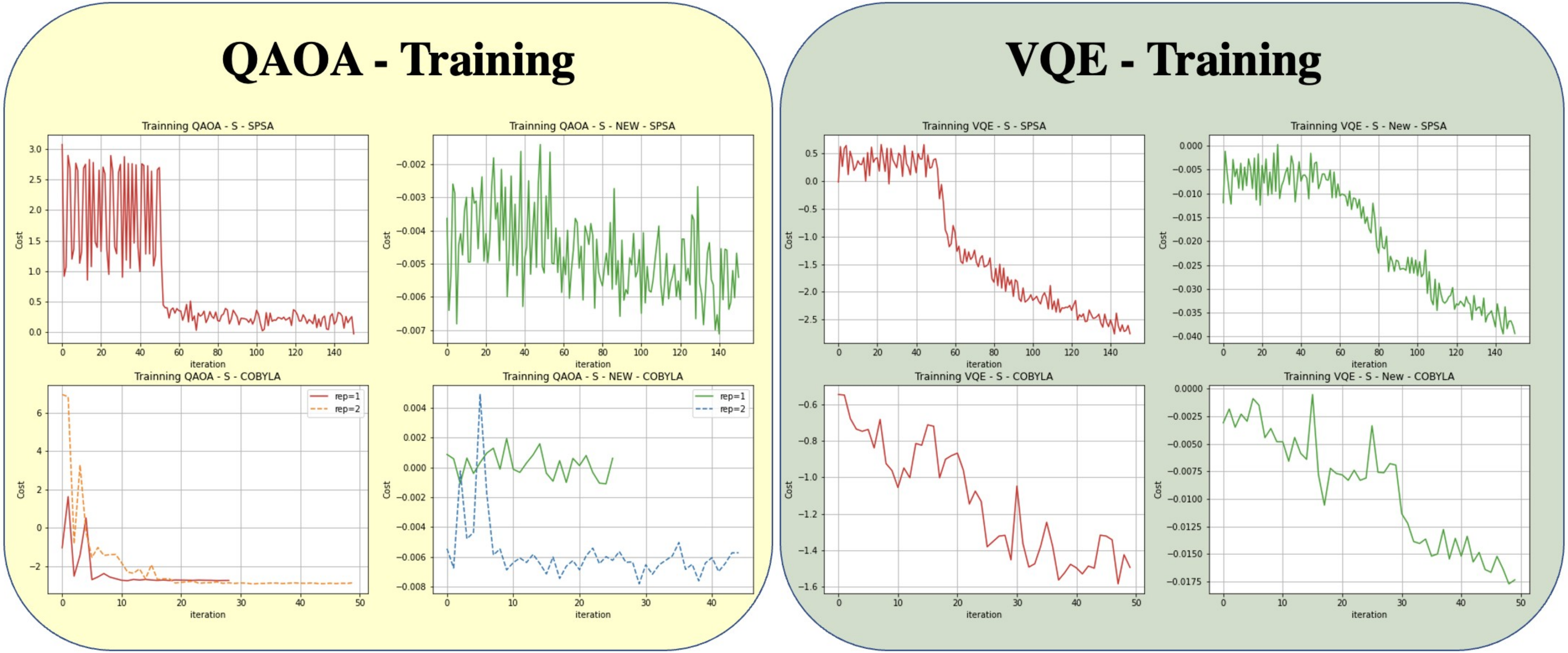
is the covariant matrix of the returns at time t and is the risk aversion

$$\Delta\omega_t = \omega_t - \omega_{t-1}$$

where is the optimal parabolic coefficient of the transaction cost

$$u_t$$

is a vector of ones with dimension N and is a Lagrange multiplier for the Budget constraint.


QAOA - Training
Trainning QAOA - S - SPSA
Trainning QAOA - S - NEW - SPSA
Trainning QAOA - S - COBYLA
Trainning QAOA - S - NEW - COBYLA
rep=1
rep=2
VQE - Training
Trainning VQE - S - SPSA
Trainning VQE - S - New - SPSA
Trainning VQE - S - COBYLA
Trainning VQE - S - New - COBYLA
Cost
iteration

XS model - 3 Stocks – 2 Periods

	Method	Solver	Cost fun	Solution	Profit [%]	Transaction Cost [%]
0	QAOA	SPSA	-0.141321	[[1.0, 0.0, 1.0], [1.0, 1.0, 0.0]]	[9.0, 5.1]	[0.1, 0.1]
1	QAOA	COBYLA	-0.141321	[[1.0, 0.0, 1.0], [1.0, 1.0, 0.0]]	[9.0, 5.1]	[0.1, 0.1]
2	VQE	COBYLA	-0.141321	[[1.0, 0.0, 1.0], [1.0, 1.0, 0.0]]	[9.0, 5.1]	[0.1, 0.1]
3	VQE	SPSA	-0.141321	[[1.0, 0.0, 1.0], [1.0, 1.0, 0.0]]	[9.0, 5.1]	[0.1, 0.1]
4	CPLEX		-0.141321	[[1.0, 0.0, 1.0], [1.0, 1.0, 0.0]]	[9.0, 5.1]	[0.1, 0.1]

S model – 5 Stocks – 3 Periods

	Method	Solver	Cost fun	Solution	Profit [%]	Transaction Cost [%]
0	QAOA	SPSA	-0.095315	[[1, 1, 0, 1, 1], [1, 1, 1, 1, 0], [0, 1, 1, 1...	[2.9, 4.8, 1.7]	[0.1, 0.05, 0.05]
1	QAOA	COBYLA	0.221688	[[1, 0, 1, 1, 1], [0, 1, 1, 1, 1], [0, 1, 1, 0...	[2.7, 4.6, 1.7]	[0.1, 0.05, 0.025]
2	VQE	COBYLA	-0.091058	[[1, 0, 1, 1, 1], [0, 1, 1, 1, 1], [0, 1, 1, 1...	[2.7, 4.6, 1.8]	[0.1, 0.05, 0.0]
3	VQE	SPSA	-0.083458	[[1, 0, 1, 1, 1], [0, 1, 1, 1, 1], [1, 1, 1, 0...	[2.7, 4.6, 1.0]	[0.1, 0.05, 0.05]
4	CPLEX		-0.095315	[[1, 1, 0, 1, 1], [1, 1, 1, 1, 0], [0, 1, 1, 1...	[2.9, 4.8, 1.7]	[0.1, 0.05, 0.05]

M model – 8 Stocks – 3 Periods

	Method	Solver	Cost fun	Solution	Profit [%]	Transaction Cost [%]
0	QAOA	SPSA	-0.154312	[[1, 1, 1, 1, 0, 1, 0, 0], [1, 1, 1, 0, 1, 1, ...	[7.4, 3.5, 4.3]	[0.1, 0.06, 0.04]
1	QAOA	COBYLA	-0.161050	[[1, 1, 1, 1, 1, 1, 1, 0], [1, 0, 1, 1, 1, 1, ...	[8.2, 4.6, 3.1]	[0.14, 0.04, 0.06]
2	CPLEX		-0.182748	[[1, 1, 1, 1, 1, 1, 1, 0], [1, 0, 1, 0, 1, 0, ...	[8.2, 5.6, 4.2]	[0.14, 0.1, 0.08]

3. Our novel Approach for the cost function

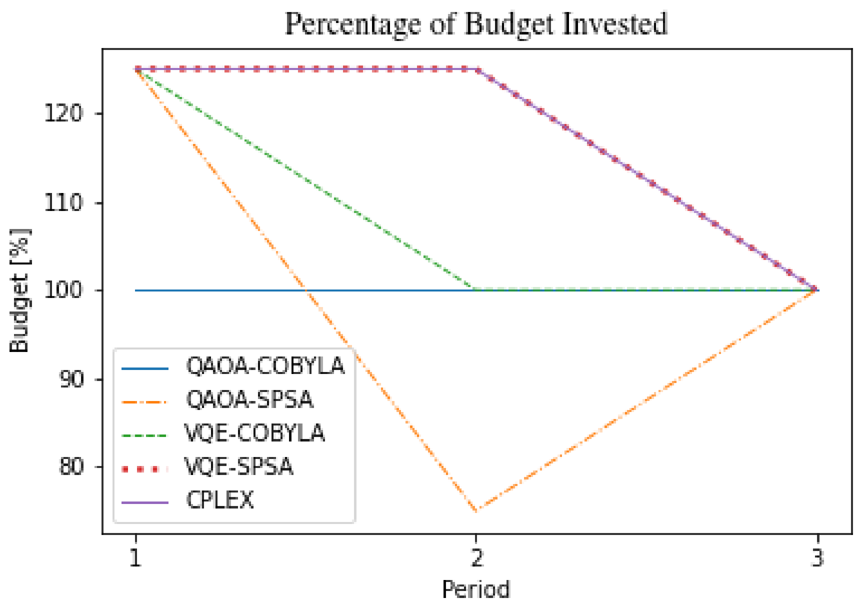
$$\mathcal{H} = -\sum_{t=t_i}^{tf} \mu_t^T \omega_t + \frac{\gamma}{2}\omega_t^T \Sigma_t \omega_t + \lambda(\Delta\omega_t)^2 + \boxed{\rho\frac{u^T\kappa}{|u^T\mu_t|}(u^T\omega_t - 1)^2} + \boxed{\beta\kappa^T\omega_t}$$

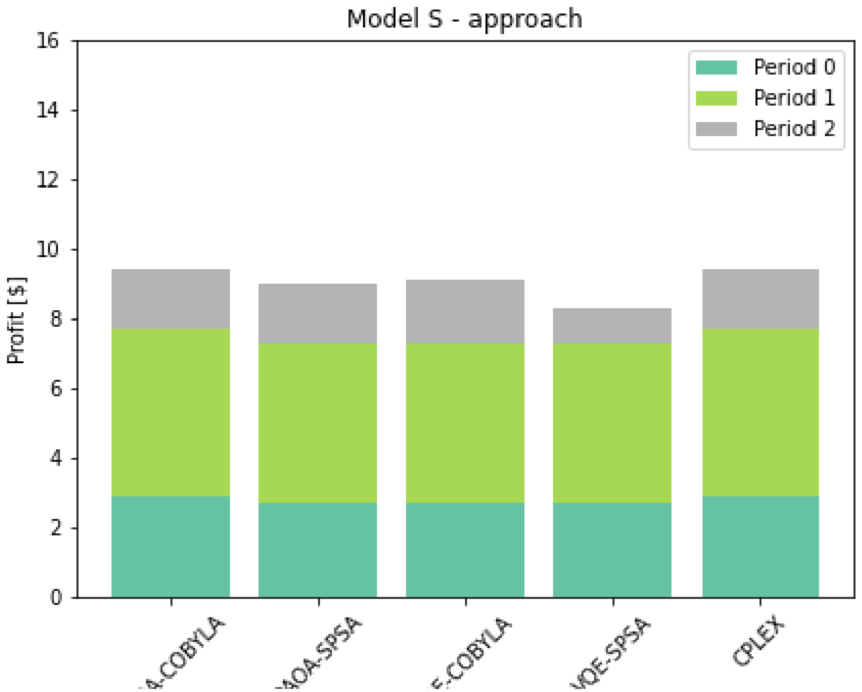
Weakly Budget Constraint

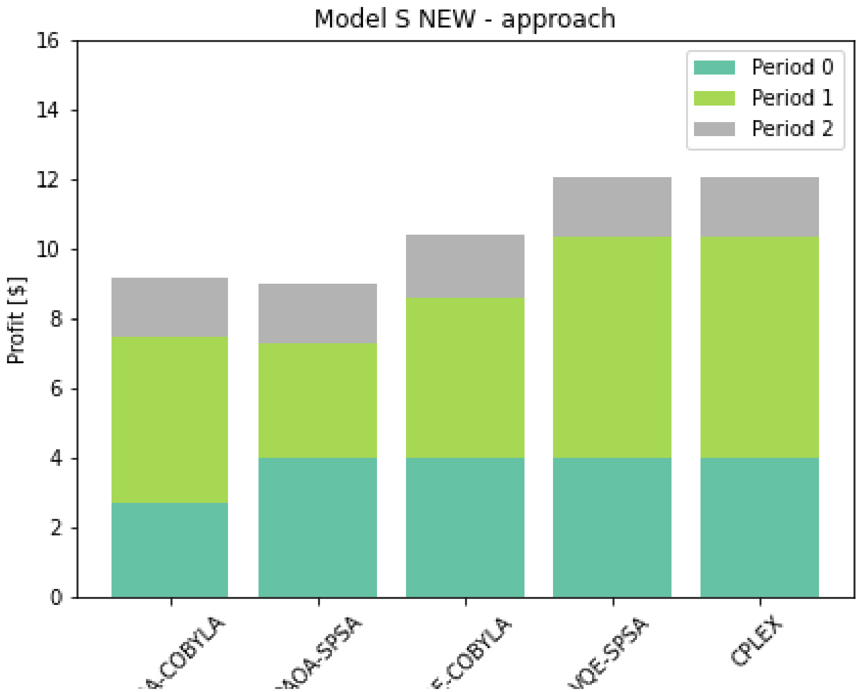
Stock forecasting uncertainty

A large sum of the values would make the constraint weak while if the forecasting mean relative error is large, it will make the constraint stronger

κ is a vector with the mean relative error of the forecasting method for the test data and is the respective Lagrange multiplier.


Percentage of Budget Invested
Budget [%]
120
110
100
90
80
QAOA-COBYLA
QAOA-SPSA
VQE-COBYLA
VQE-SPSA
CPLEX
1
2
3
Period


Model S - approach
Profit [$]
16
14
12
10
8
6
4
2
0
Period 0
Period 1
Period 2
VQE-SPSA
CPLEX


Model S NEW - approach
Profit [$]
16
14
12
10
8
6
4
2
0
Period 0
Period 1
Period 2
VQE-SPSA
CPLEX

Conclusion and Future Work

- We have come with a QNN model capable of forecasting the price trend for different assets. This model present some advantages when compare with classical approaches.
- We implement satisfactorily the problem of optimization portfolio using qiskit with two quantum solvers QAOA and VQE, and we compare the results with a classical solver CPLEX. Even though we select a small number of maximal iterations, the quantum models come to the optimal solution.
- We implement a new approach for the objective function called the budget increment opportunity, where if there is a great opportunity of investment (high bare return and low uncertainty in the forecasting) the budget constraint becomes weak. This approach allows us to get a considerably increment in profits.
- For next work, we want to implement these methods on a real hardware. Unfortunately, we couldn’t make it because some technical difficulties with the two backends where we tried it. Additionally, we want to add fundamental analysis as input to the QNN, to explore new ways of improving the forecasting ability.