Time Series Forecasting with Quantum Machine Learning Architectures

Chapter · October 2022

DOI: 10.1007/978-3-031-19493-1_6

3 authors, including:

Mayra Alejandra Rivera-Ruiz
Center for Research and Advanced Studies of the National Polytechnic Institute
4 PUBLICATIONS **15** CITATIONS
SEE PROFILE

Andres Mendez-Vazquez
Center for Research and Advanced Studies of the National Polytechnic Institute
39 PUBLICATIONS **206** CITATIONS
SEE PROFILE

Time Series Forecasting with Quantum Machine Learning Architectures

Mayra Alejandra Rivera-Ruiz[1(✉)], Andres Mendez-Vazquez[1], and José Mauricio López-Romero[2]

[1] CINVESTAV Unidad Guadalajara, Av. del Bosque 1145, colonia el Bajío, C.P, 45019 Zapopan, Jalisco, Mexico
{mayra.rivera,andres.mendez}@cinvestav.mx

[2] CINVESTAV Unidad Querétaro, Libramiento Norponiente 2000, Fracc. Real de Juriquilla, C.P, 76230 Santiago de Querétaro, Querétaro, México
jm.lopez@cinvestav.mx

Abstract. Time series forecasting has been a topic of special interest due to its applications in Finance, Physics, Environmental Sciences and many other fields. In this article, we propose two classical-quantum hybrid architectures for time series forecasting with a multilayered structure inspired by the multilayer perceptron (MLP): a Quantum Neural Network (QNN) and a Hybrid-Quantum Neural Network (HQNN). These architectures incorporate quantum variational circuits with specific encoding schemes and the optimization is carried out by a classical computer. The performance of the proposed hybrid models is evaluated in four forecasting problems: Mackey-Glass time series and USD-to-euro currency exchange rate forecasting (univariate time series) as well as Lorenz attractor and prediction of the Box-Jenkins (Gas Furnace) time series (multivariate time series). The experiments were conducted by using the built-in Pennylane simulator lighting.qubit and Pytorch. Finally, these architectures, compared to the MLP, CNN and LSTM show a competitive performance with a similar number of trainable parameters.

Keywords: Quantum machine learning · Time series forecasting · Quantum neural networks

1 Introduction

A time series is a series of data ordered in time. There are two types of time series: multivariate time series and univariate time series [4]. The former involves multiple dependent variables whereas in the latter there is only one dependent variable. Time series forecasting aims to find the dependency between the past values and the future values of the time series. Time series has been an active research topic and can be found in areas of special interest such as economy, finance, and physical sciences [6,23].

O. Pichardo Lagunas et al. (Eds.): MICAI 2022, LNAI 13612, pp. 66–82, 2022.
https://doi.org/10.1007/978-3-031-19493-1_6

In the last few years, machine learning has become quite popular in a wide range of areas including time series forecasting. Specifically, there have been many attempts to use Artificial Neural Networks (ANNs) as an alternative to traditional statistical methods due to the fact that they are universal function approximators [31]. Additionally, Machine Learning models have been compared to traditional statistical methods, showing that neural networks are generally more accurate, and in specific situations, they improve significantly the accuracy of time series data prediction [9,15,18]. Specifically, Recurrent Neural Networks (RNNs) are widely used to solve time series forecasting due to their success in modeling sequential data [14]. Long Short-Term Memory Networks (LSTM) aim to solve the vanishing gradient problem of standard RNNs and are capable of learning long-term dependencies, which makes them very efficient for time series forecasting [10]. Quantum versions of RNNs and LSTM are proposed in [5,11]. On the other hand, a Convolutional Neural Network (CNN) is a class of ANN mainly used for computer vision and image processing tasks. Recently, CNNs have been applied for time series forecasting with very promising results [3,24,30]. A quantum version of CNN is introduced in [13] and has been applied in image classification [2,17,27].

On the other hand, Quantum Computing promises improvements in solving complex problems that the most powerful supercomputers cannot solve. Thus, Quantum Machine Learning (QML) has emerged as a tool to find patterns in data [8,16]. Specifically, Variational Quantum Circuits (VQCs) have emerged as a quantum counterpart of Neural Networks [21]. VQCs have begun to be used as a possible alternative to Deep Learning and are known in the literature as Quantum Neural Network or Quantum Circuit Learning (QCL). In this framework entangling operations are used to create multilayered structures, similar to Deep Learning. Furthermore, QCL is a classical-quantum hybrid algorithm that can perform supervised learning tasks [21]. In QCL, the output is computed by a quantum circuit whereas the update of parameters is performed by a classical computer. Since quantum circuits admit a universal model [7], quantum hybrid architectures become attractive for time series forecasting.

Classical models inspired by quantum computing have been proposed for predicting chaotic time series produced by a Lorentz system with competitive results when compared with classical neural networks [28,29]. On the other hand, quantum models have been proposed and evaluated with univariate time series showing to be on par with existing classical models [19,25,26]. In [19] stock prices using a quantum reservoir computer are predicted. Models based on a quantum optimization algorithm are proposed in [25,26]. In [26] a new neutrosophic set based time series forecasting model is presented, where a quantum optimization algorithm is used to improve the accuracy. The model is verified and validated with datasets of the university enrollment of Alabama (USA), Taiwan futures exchange (TAIFEX) index and Taiwan Stock Exchange Corporation (TSEC) weighted index. In [25] a fuzzy-quantum time series forecasting model (FQTSFM) is introduced. The FQTSFM is tested with three different datasets, namely the daily average temperatures of Taipei, TAIFEX index and

TSEC weighted index. Otherwise, QML is applied for finance in [12], where a parameterized quantum circuit (PQC) is used for forecasting time series signals with simulated quantum forward propagation.

In this paper the contribution is threefold:

- A Quantum Neural Network (QNN) for time series forecasting that consists of a multilayered VQC is proposed, where the number of qubits is set according to the size of the regressor vector.
- A Hybrid-Quantum Neural Network (HQNN) for time series forecasting is presented, where two or more dressed quantum circuits (quantum circuits sandwiched between two classical layers) are concatenated in order to construct a multilayered architecture.
- The range of applicability of previous quantum models [19,25,26] is extended by including not only univariate but also multivariate time series forecasting.

From the variety of experiments performed in this work, it is possible to conclude that these hybrid models show a competitive performance when compared to MLP, CNN and LSTM in univariate and multivariate time series forecasting.

The rest of the paper is organized as follows. The basics of quantum computing are covered in Sect. 2. In Sect. 3, the problem to be solved is established. The proposed architectures are formalized in Sect. 4. The experimental setups and results are presented in Sect. 5. Finally, the conclusions are presented in Sect. 6.

2 Preliminaries

We begin by introducing basic concepts of quantum computing before presenting our contribution. In this section we use the notation introduced in [20].

2.1 The Qubit

The qubit is the basic unit of information in quantum computing. A qubit is a two-state quantum mechanical system. This means that while a classical bit can be 0 or 1, a qubit can also exist in what we call a superposition state, which is a linear combination of the states $|0\rangle$ and $|1\rangle$. This property is the main difference between the bit and the qubit [22]. Mathematically, we can represent a qubit as a two-dimensional complex vector in which the basis vectors are the allowed states $|0\rangle$ and $|1\rangle$. These vectors are given by $|0\rangle = \binom{1}{0}$ and $|1\rangle = \binom{0}{1}$. Then a superposition state is written as

$$|\Psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle\,, \tag{1}$$

where α and β are complex numbers that satisfy the normalization condition: $|\alpha|^2 + |\beta|^2 = 1$. We can construct multi-qubit states through the tensor product. For example, the state $|01\rangle = |0\rangle \otimes |1\rangle$ represents a two-qubit system where the first and second qubit are in the state $|0\rangle$ and $|1\rangle$, respectively.

The state of the qubits could be modified through unitary operations (quantum gates) to perform a computation. Quantum circuits are used for this purpose. The final state of a quantum circuit is the measurement of the qubits in a given basis.

2.2 Quantum Circuits

A quantum circuit is a model of quantum computation that consists of a sequence of quantum operations (quantum gates) applied to quantum data (multi-qubit states). A quantum gate is a unitary operator which can be described as a unitary matrix. Then, if we have a set of unitary operators $\{U_i\}_{i=1}^{k}$, we can mathematically represent the state $|\Psi\rangle$ prepared by the quantum circuit as

$$|\Psi\rangle = \prod_{i=1}^{k} U_i \, |0\rangle^{\otimes n}, \tag{2}$$

where $|0\rangle^{\otimes n}$ is an initial n-qubit state. The graphical representation of Eq. 2 is shown in Fig. 1.

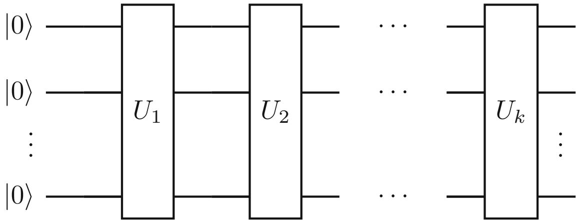


Fig. 1. Graphical depiction of the quantum circuit of $\prod_{i=1}^{k} U_i \, |0\rangle^{\otimes n}$

The quantum gates can act on one (single-qubit gate) or several qubits (multiple-qubit gates). The quantum NOT gate can be represented with the Pauli-X matrix. The NOT operator acts on a single qubit, and it is equivalent to a NOT classical gate. Hence, we have: $U_{\text{NOT}} |0\rangle = |1\rangle$ and $U_{\text{NOT}} |1\rangle = |0\rangle$. Since the other Pauli matrices are unitary, they are also used as single-qubit gates. The action of the Pauli Y and Z gates is a rotation around the Y-axis and Z-axis respectively of the Bloch sphere by π radians. Quantum gates can also be parametrised. The main parametrised gates are the Pauli rotations: $R_x = e^{-i\frac{\theta}{2}X}$, $R_y = e^{-i\frac{\theta}{2}Y}$ and $R_z = e^{-i\frac{\theta}{2}Z}$, which rotate the Bloch vector about the x, y and z axes by an angle θ.

The quantum gate that allows us to create superposition states is known as Hadamard gate. It acts on a single qubit and maps the basis states $\{|0\rangle, |1\rangle\}$ into the superposition states $\left\{\frac{|0\rangle+|1\rangle}{\sqrt{2}}, \frac{|0\rangle+|1\rangle}{\sqrt{2}}\right\}$. Another important quantum gate is the CNOT gate and acts on a qubit in the following way: $|a, b\rangle \rightarrow |a, b \oplus a\rangle$.

Mathematically serial operations are represented by a matrix product starting with the first operation on the left followed by subsequent operations moving to the right. On the other hand, when quantum operations are performed in parallel, we need to compute the tensor product.

2.3 Variational Quantum Circuits

A Variational Quantum Circuit (VQC) is a quantum circuit with adjustable parameters which are optimized in a classical computer. VQCs can be used as a quantum version of classical feed-forward neural networks [1,21].

A VQC consists of three parts:

- **Embedding circuit:** The classical data $\mathbf{x} = (x_1, ..., x_N)$ is encoded into a quantum state by applying a $\mathbf{x}$-parameterized unitary operator $U_{\text{in}}(\mathbf{x})$ to an initial quantum state $|0\rangle^{\otimes n}$:

$$\mathcal{E} : \mathbf{x} \rightarrow |\Psi_{\text{in}}(\mathbf{x})\rangle = U_{\text{in}}(\mathbf{x})\,|0\rangle^{\otimes n}. \tag{3}$$

  In other words, $\mathcal{E}$ maps the classical input from a classical vector space to a Hilbert space. A common choice of $U_{\text{in}}(\mathbf{x})$ is to apply a Pauli rotation gate (e.g., a Pauli-Y rotation gate), for each input feature in $\mathbf{x}$ as follows: $U_{\text{in}}(\mathbf{x}) = R_y(x_1) \otimes \cdots \otimes R_y(x_N) = \otimes_{i=1}^{N} R_y(x_i)$.
- **Parameterized circuit**: A parameterized circuit of depth q is equivalent to the hidden layers of the ANNs and is a concatenation of quantum layers:

$$\mathcal{U} = \mathcal{L}_q \circ \cdots \mathcal{L}_2 \circ \mathcal{L}_1, \tag{4}$$

  where $\mathcal{U}$ is a unitary operator. A quantum layer $\mathcal{L}$ consists of single-qubit operations followed by entangling operations (multi-qubit operations).
- **Measurement:** To obtain the output vector $\mathbf{y}$, the expectation values of the n observables $\hat{\mathbf{y}} = [\hat{y}_1, ..., \hat{y}_n]$ are measured. The observable $\hat{y}_i$ is a tensor product of n operators as follows:

$$\begin{aligned} \hat{y}_1 &= B \otimes I \otimes I ... \otimes I \\ \hat{y}_2 &= I \otimes B \otimes I ... \otimes I \\ &\vdots \\ \hat{y}_n &= I \otimes I \otimes I ... \otimes B \end{aligned}$$

  where I is the identity operator and B is a Hermitian operator that could be different for each observable. Usually, B is chosen as the Pauli-Z operator. This operation is defined as a measurement layer:

$$\mathcal{M} : |\Psi_{\text{out}}\rangle \rightarrow \mathbf{y} = \langle\Psi_{\text{out}}|\,\hat{\mathbf{y}}\,|\Psi_{\text{out}}\rangle, \tag{5}$$

  where $|\Psi_{\text{out}}\rangle$ is the transformed qubit state. $\mathcal{M}$ is a map from a quantum state to a classical vector.

Finally, the full quantum network is given by

$$Q = \mathcal{M} \circ \mathcal{U} \circ \mathcal{E}. \tag{6}$$

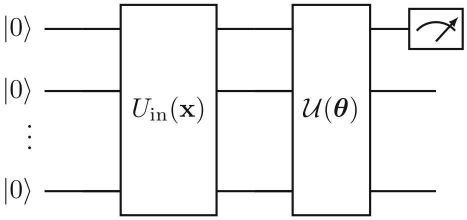

$|0\rangle$
$|0\rangle$
$|0\rangle$
$U_{\text{in}}(\mathbf{x})$
$\mathcal{U}(\boldsymbol{\theta})$

Fig. 2. Illustration of a variational quantum circuit. The input $\mathbf{x}$ is encoded into a quantum state by the embedding circuit. $\mathcal{U}(\boldsymbol{\theta})$ is the parameterized circuit with the trainable parameters $\boldsymbol{\theta}$ of the network. A quantum measurement is made on the first qubit.

3 Problem Statement

A time series is a sequence of observations of the values that one (univariate time series) or more variables (multivariate time series) take over time, where the time interval between observations is constant. For a time series of size N, the main problem is to predict $x(t + \Delta t)$ based on the input:

$$\mathbf{x} = (x(t + \Delta t - P), ..., x(t + \Delta t - 2), x(t + \Delta t - 1)), \tag{7}$$

where $P < N$. In other words, the objective is to predict $x(t + \Delta t)$ using P past values building the forecasting model $\hat{x}(t + \Delta t) = f(\mathbf{x})$. This problem can be approached as a supervised learning problem.

The supervised learning scheme applied to time series will have as input the regressor vector $\mathbf{x}$ given by Eq. 7 and the corresponding teacher data will be given by the future values $x(t + \Delta t)$.

4 Quantum Machine Learning Architectures for Time Series Forecasting

In this section, the two proposed architectures for time series forecasting are described.

4.1 Quantum Neural Network

As we mentioned earlier, one way to construct a quantum counterpart of neural networks is in terms of VQCs. A general structure of a VQC is shown in Fig. 2. Thus, the output is obtained from a quantum device (quantum circuit) whereas the optimization of parameters $\boldsymbol{\theta}$ is performed by a classical device (classical computer). To this end, the regressor vector $\mathbf{x}$ is encoded into a quantum state as in Eq. 3, where the number of qubits n is equal to the P past values.

Thus, selecting the Pauli-Y rotation gate, the input quantum state can be expressed as:

$$|\Psi_{\text{in}}(\mathbf{x})\rangle = \bigotimes_{j=1}^{n} R_Y(\pi \mathbf{x}_j)|0\rangle^{\otimes n}. \tag{8}$$

The output state $|\Psi_{\text{out}}(\mathbf{x}, \boldsymbol{\theta})\rangle$ can be obtained by applying a $\boldsymbol{\theta}$-parametrized unitary $\mathcal{U}(\boldsymbol{\theta})$ to the input quantum state:

$$|\Psi_{\text{out}}(\mathbf{x}, \boldsymbol{\theta})\rangle = \mathcal{U}(\boldsymbol{\theta})|\Psi_{\text{in}}(\mathbf{x})\rangle. \tag{9}$$

The $\boldsymbol{\theta}$-parametrized unitary $\mathcal{U}(\boldsymbol{\theta})$ consists of l layers. The k-th layer is chosen to be

$$\mathcal{L}^{(k)}(\theta_k) = U_{\text{ent}} \bigotimes_{j=1}^{n} R_Z(\theta_{j,k}^{Z}) R_Y(\theta_{j,k}^{Y}) R_X(\theta_{j,k}^{X}), \tag{10}$$

where the entangling gate U_{ent} is composed of CNOT gates. Consequently, $\mathcal{U}(\boldsymbol{\theta})$ can be expressed in compact form as follows:

$$\mathcal{U}(\boldsymbol{\theta}) = \mathcal{L}^{(D)}(\theta_D)\mathcal{L}^{(D-1)}(\theta_{D-1}) \cdots \mathcal{L}^{(2)}(\theta_2)\mathcal{L}^{(1)}(\theta_1). \tag{11}$$

The proposed VQC is shown in Fig. 3.

Then, in the simplest case of one-step-ahead forecasting models, the estimated output $\hat{x}(t + \Delta t)$ can be expressed as the following expectation value:

$$\hat{x}(t + \Delta t) = \langle \Psi_{\text{out}}|B|\Psi_{\text{out}}\rangle, \tag{12}$$

were B is some chosen observable. The loss function L corresponds to the conventional squared loss

$$L = || \; x(t + \Delta t) - \hat{x}(t + \Delta t) \; ||^2, \tag{13}$$

due to the fact that time series forecasting is approached as a regression problem.

4.2 Hybrid Quantum Neural Network

In the previous subsection, we used a VQC as a model for time series forecasting with a multilayered structure. However, one of its limitations is that the number of features depend on the number of qubits, contrary to classical feed-forward neural networks, where we can freely choose the number of features of each layer. One way of overcoming this constraint is by sandwiching the VQC between two classical layers as shown in Fig. 4. This solution is formalized by introducing the concept of Dressed Quantum Circuit (DQC) [20]. The general form of a DQC is given by:

$$\tilde{Q}_{n_{\text{in}} \to n_{\text{out}}} = L_{n_q \to n_{\text{out}}} \circ Q_{n_q \to n_q} \circ L_{n_{\text{in}} \to n_q}, \tag{14}$$

where $L_{n \to n'}$ is a classical layer with n input and n' output variables. $Q_{n_q \to n_q}$is the quantum circuit given by Eq. (6). The classical layer $L_{n_q \to n_{\text{out}}}$ is responsible of the pre-processing of the inputs of the VQC whereas $L_{n_{\text{in}} \to n_q}$ performs the

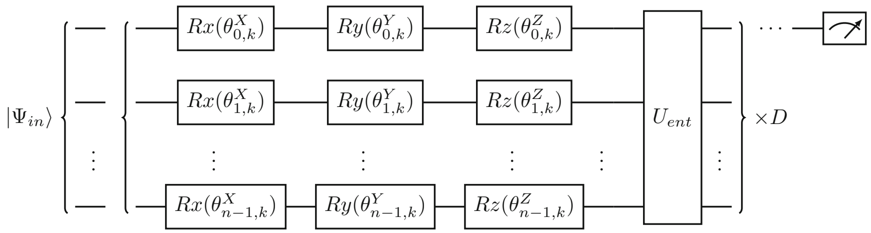


Fig. 3. VQC used for the proposed model. $|\Psi_{\text{in}}\rangle$ is the input state prepared by the input gate $U_{\text{in}}(\mathbf{x})$. D denotes the depth of the circuit. The U_{ent} gate is composed of CNOT gates.

post-processing of the outputs of the VQC [20]. Since both classical layers contain trainable parameters, the pre-processing of inputs and the post-processing of outputs of the VQC are optimized during the training process. The main advantage of a DQC is that n_{in} and n_{out} (the number of input and output variables) are independent of the number of qubits of the VQC.

Inspired by the Classical Deep Neural Networks we propose a multilayered hybrid classical-quantum neural network (HQNN):

$$\text{HQNN} = \tilde{Q}_{n_{d-1}\to n_d} \circ \cdots \tilde{Q}_{n_1\to n_2} \circ \tilde{Q}_{n_0\to n_1}, \tag{15}$$

where d denotes the depth of the HQNN.

In this work, a two-layer HQNN is applied to time series forecasting:

$$\text{HQNN} = \tilde{Q}_{n_1\to n_2} \circ \tilde{Q}_{n_0\to n_1}, \tag{16}$$

where n_0 is equal to the number of variables of the regressor vector. In the case of one-step ahead forecasting $n_2 = 1$. We use the same structure for the DQC of each layer. Then, the encoding and variational layers of the l-th layer of the HQNN are given by:

$$\mathcal{E}_l(\mathbf{x}) = \bigotimes_{j=1}^{n} R_Y\left(\tanh\left(x_{j,l}\right)\right)|0\rangle^{\otimes n}, \tag{17}$$

$$\mathcal{U}_l\left(\boldsymbol{\theta}\right) : |x\rangle \longrightarrow |y\rangle = U_{\text{ent}} \bigotimes_{j=1}^{n} R_Z(\theta^Z_{j,l}) R_Y(\theta^Y_{j,l}) R_X(\theta^X_{j,l}), \tag{18}$$

where U_{ent} is composed of CNOT gates. In the measurement layer we take the expectation value of the Pauli-Z operator of each qubit.

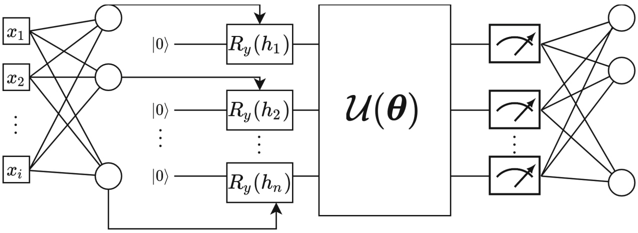


Fig. 4. Illustration of a dressed quantum circuit. It is built with a VQC between two classical layers. The k-th output from the classical pre-processing layer is denoted by h_k.

5 Results and Discussion

In this section, we evaluate the performance of the proposed quantum machine learning architectures by applying the models to four standard problems: Mackey-Glass time series, Lorenz attractor, prediction of the Box-Jenkins time series and USD-to-euro currency exchange rate forecasting.

The experiments performed in this research are conducted by using the built-in Pennylane simulator lighting.qubit and Pytorch. Additionally, for the purpose of a fair comparison, the same optimizers, learning rates and number of epochs (100 for all cases) are applied to the MLP, CNN, LSTM and the proposed models. In addition, the time series data is scaled within the range $[0, 1]$ using the min-max normalization formula.

The SGD optimizer is used in the Lorenz attractor problem with a learning rate of 0.001. In the case of Mackey-Glass and USD-to-euro currency exchange, the Adam Optimizer is used with a learning rate of 0.001. In the case of the Box-Jenkins problem, the Adam Optimizer is used with a learning rate of 0.0005.

The depth of the QNN and the number of qubits used in the HQNN are equal to 5 and 3 in the Mackey-Glass time series problem, 7 and 4 for the Lorenz attractor, 10 and 2 in the Box-Jenkins time series and finally, 4 and 3 in the USD to Euro currency exchange rate forecasting.

The CNN model consists of a 1D convolutional layer and two fully connected layers. After the 1D convolutional layer, a 1D max-pooling layer is applied. The ReLU activation function is used in the convolutional layer and the first fully-connected layer. In the case of the Lorenz attractor problem, the number of output channels of the convolutional layer and the number of units in the first fully connected layer is equal to 8 and 16, respectively, and in the rest of the time series datasets, equal to 4 and 8. The LSTM model has a hidden state size equal to 5 in the Lorenz attractor problem and equal to 3 in the rest of the time series datasets.

The performance results of the proposed models in terms of RMSE, MAE and MAPE are summarized in Tables 1, 2, 3 and 4.

Mackey-Glass Time Series. Mackey-Glass time series data are derived from the following differential equation:

$$\dot{x} = \frac{\alpha x(t-\tau)}{1 + x^{\gamma}(t-\tau)} - \beta x(t), \tag{19}$$

where τ is the time-delay parameter. When $\tau \geqslant 17$ Eq. (19) shows the chaotic phenomenon. To obtain its numerical solution, the Runge-Kutta method is applied with the initial condition $x(0) = 1.2$ and integration step equal to 0.1. The parameters α, β and γ are equal to 0.2, 0.1 and 10, respectively. We use 1000 simulation data points to build our model, which are described as follows:

$$[x(t-18), x(t-12), x(t-6), x(t); x(t+6)], \tag{20}$$

where $t = 19, 20, ..., 1017, 1018$. The first 500 points are selected as the training data and the rest as the testing data. The vector $\mathbf{x} = (x(t-18), x(t-12), x(t-6), x(t))$ is selected as the input vector and the last variable $x(t+6)$ as the output variable of the model. Figures 5a and 6a show the comparison between the original and forecasted outputs of both models QNN as well as HQNN for the training and validation phases. Additionally, the Fig. 7a shows the MSE loss curves of the HQNN, QNN, MLP, CNN and LSTM. When comparing the QNN and MLP, the QNN shows a superior performance. We can observe an improvement of about 34% in RMSE, 33% in MAE, and 33% in MAPE. On the other hand, the MLP shows a better performance than the HQNN, improving the RMSE, MAE and MAPE by approximately 3%, 12% and 1%, respectively. Additionally, the QNN outperforms the LSTM. We can observe an improvement of about 27% in RMSE, 18% in MAE and 20% in MAPE.

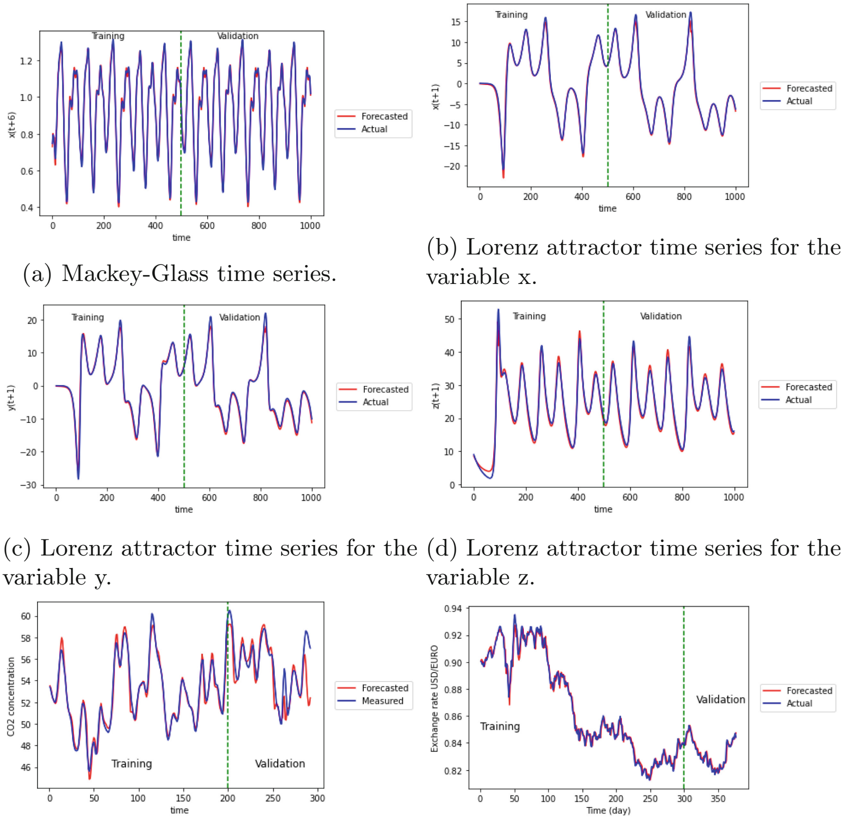
Table 1. Comparison results among the proposed models, MLP, CNN and LSTM for Mackey-Glass time series prediction.

Network	Parameters	RMSE	MAE	MAPE
QNN	60	0.01360	0.0114	0.01319
HQNN	68	0.02117	0.01726	0.01992
MLP (4,5,6,1)	68	0.02047	0.01697	0.01963
CNN	65	0.04586	0.03419	0.04159
LSTM	76	0.01866	0.01395	0.01659

Lorenz Time Series. The Lorenz equations are given by

$$\begin{aligned} \dot{x} =& \sigma(y - x), \\ \dot{y} =& -y - zx + \rho x, \\ \dot{z} =& -\beta z + xy. \end{aligned} \tag{21}$$

Let $\sigma = 10$, $\rho = 28$ and $\beta = 8/3$. We get its numerical solutions by using the Euler method to obtain the time series dataset by taking the initial conditions:



(a) Mackey-Glass time series.

(b) Lorenz attractor time series for the variable x.

(c) Lorenz attractor time series for the variable y.

(d) Lorenz attractor time series for the variable z.

(e) Box-Jenkins Gas Furnace Data.

(f) Exchange Rate USD/EUR.

Fig. 5. Actual and forecasted outputs using the QNN.

$x(0) = 0$, $y(0) = -0.01$ and $z(0) = 9$. The vector $\mathbf{x} = [x(t), y(t), z(t)]$ is chosen as the input vector and the three variables $x(t+1), y(t+1), z(t+1)$ are the output variables, where $t = 1, ..., 1000$. The first 500 points are used during the training phase and the rest in the testing phase.

For the QNN we use a different unitary input gate:

$$U_{\text{in}} = \bigotimes_{j=0}^{n-1} R_Y(\cos^{-1}(\mathbf{x}_j^2))R_Z(\sin^{-1}(\mathbf{x}_j)). \tag{22}$$

To get the output variables, we take the following expectation values:

$$\begin{aligned}\hat{x}(t+1) =& \langle \Psi_{out} | X \otimes I \otimes I | \Psi_{out} \rangle,\\ \hat{y}(t+1) =& \langle \Psi_{out} | I \otimes Y \otimes I | \Psi_{out} \rangle,\\ \hat{z}(t+1) =& \langle \Psi_{out} | I \otimes I \otimes Z | \Psi_{out} \rangle.\end{aligned} \tag{23}$$

Figures 5b–5d and 6b–6d show the comparison between the original and forecasted outputs of both models QNN as well as HQNN for $x(t+1)$, $y(t+1)$ and $z(t+1)$ in the training and validation phases. In addition, the Fig. 7b shows the MSE loss curves of the HQNN, QNN, MLP, CNN and LSTM. When comparing the QNN, HQNN and MLP, the HQNN shows the worst performance when predicting $y(t+1)$ and $z(t+1)$ despite the fact that this model has more trainable parameters. When forecasting $x(t+1)$ the HQNN slightly outperformed the MLP, improving the RMSE and MAE by about 7% and 3%, respectively. On the other hand, when forecasting $x(t+1)$, the QNN noticeably outperformed the MLP. We can observe an improvement of approximately 97% in RMSE, 75% in MAE, and 77% in MAPE. Similarly, for $y(t+1)$, the QNN outperformed the MLP, improving the RMSE, MAE and MAPE by about 51%, 52% and 81%, respectively. However, when predicting $z(t+1)$, the MLP is superior to the QNN, improving the RMSE by around 14%, the MAE by about 27% and the MAPE by approximately 26%. Additionally, the QNN outperformed the CNN in predicting $x(t+1)$, $y(t+1)$ and $z(t+1)$. When comparing the QNN and the LSTM, the QNN shows a superior performance in predicting $x(t+1)$ and $y(t+1)$. However, when predicting $z(t+1)$, the LSTM is superior to the QNN, improving the RMSE by around 5%, the MAE by about 15% and the MAPE by approximately 28%.

Box-Jenkins Gas Furnace Data. The Box-Jenkins gas furnace time series is widely used in the literature. To obtain this times series, the carbon dioxide concentration (CO_2) is measured in the gases product of a combustion process of a methane-air mixture. Every 9 s a record is taken by keeping the gas flow rate constant and methane rate changing randomly. Thus, in this work, the following prediction model is chosen:

$$[\nu(t-4), y(t-1); y(t)] \qquad t = 5, ..., 296 \tag{24}$$

where $\nu(t)$ is the methane gas flow, and $y(t)$ is the CO_2 concentration in the output gases. The first 200 points are selected as the training data, and the rest as the testing data.

The comparison between the measured and forecasted outputs for the training and validation phases is shown in Figs. 5e and 6e. Furthermore, the Fig. 7c shows the MSE loss curves of the HQNN, QNN, MLP, CNN and LSTM. We can observe that the HQNN model provided better results than the QNN and MLP. The HQNN outperformed the MLP by approximately 8% in RMSE and 4% in both MAE as well as MAPE. Similarly, the HQNN is superior to the QNN, improving the RMSE by approximately 57% and the MAE and MAPE by about 51%. In addition, the HQNN outperformed the CNN, improving the RMSE by approximately 13% and the MAE and MAPE by about 12%. However, the LSTM slightly outperformed the HQNN, improving the MAE by approximately 8% and the RMSE and MAPE by about 7%.

Table 2. Comparison results among the proposed models, MLP, CNN and LSTM for the Lorenz attractor.

Network	Parameters	RMSE	MAE	MAPE
QNN	63	0.065382	0.37843	0.12376
		0.91871	0.65624	0.13911
		1.10889	0.94987	0.04039
HQNN	194	1.9695	1.44324	0.55534
		3.16348	2.24749	1.12748
		2.25607	1.72386	0.07503
MLP (3,12,9,3)	195	2.11768	1.48587	0.52687
		1.87211	1.35788	0.72274
		0.9579	0.69229	0.02976
CNN	227	2.04956	1.44207	0.40423
		2.71476	2.10219	0.85086
		2.48987	1.62512	0.0626
LSTM	178	1.95701	1.53737	0.52913
		3.22944	2.13337	1.18607
		1.05513	0.81134	0.02923

Exchange Rate USD/EUR. The data about exchange rate USD/EUR are retrieved from the site http://fx.sauder.ubc.ca/data.html. Data are gathered for the period from 01/01/2020 until 08/07/2021 with daily step. We use 376 simulation data points to build the model:

$$[x(t-4), x(t-3), x(t-2), x(t-1), x(t); x(t+1)], \tag{25}$$

where $t = 5, ..., 380$. The first 300 data points are used in the training phase and the remaining data in the testing phase.

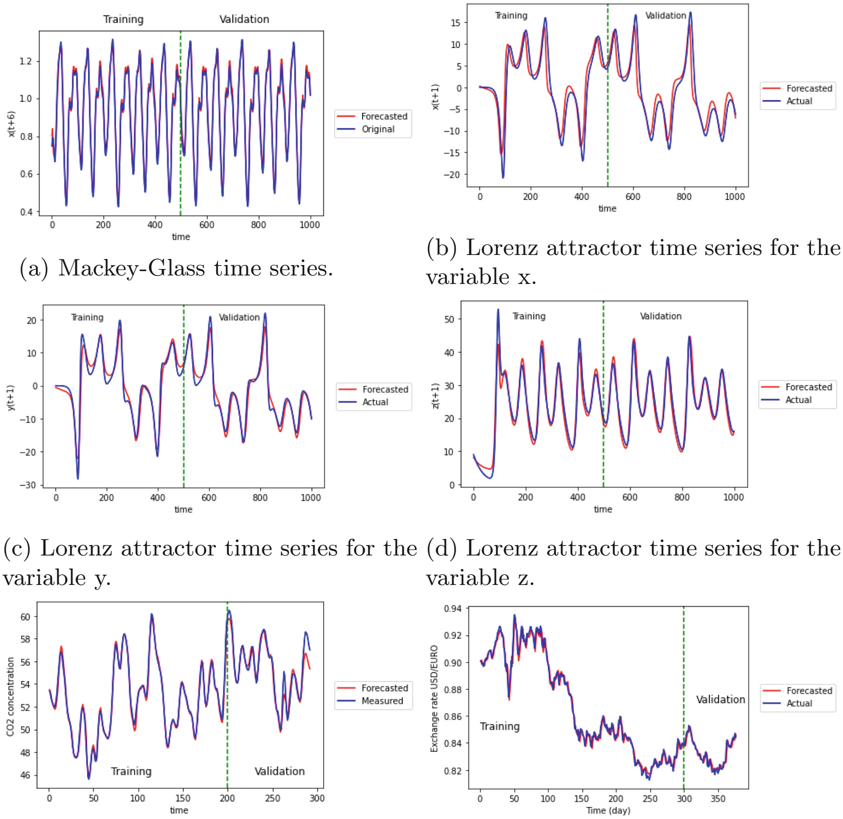
Figures 5f and 6f show the comparison between the original and forecasted outputs of both models QNN as well as HQNN for the training and validation phases. Additionally, the Fig. 7d shows the MSE loss curves of the HQNN, QNN, MLP, CNN and LSTM. When comparing the QNN, HQNN and MLP, the QNN shows the best performance of the three models. The QNN outperformed the MLP. We can observe an improvement of around 39% in RMSE, MAE, and MAPE. Similarly, the HQNN shows a better performance than the MLP, improving the RMSE by about 35% and both MAE as well as MAPE by approximately 36%. In addition, both the QNN and the HQNN outperformed the CNN and LSTM models. When comparing the QNN and CNN, the QNN showed an improvement of approximately 23% in RMSE and 22% in MAE as well as MAPE.

Table 3. Comparison results among the proposed models, MLP, CNN and LSTM for Box Jenkins time series prediction.

Network	Parameters	RMSE	MAE	MAPE
QNN	60	1.62488	0.97698	0.01733
HQNN	58	0.69741	0.47627	0.00851
MLP (2,6,5,1)	59	0.75594	0.49578	0.00885
CNN	65	0.80361	0.54213	0.00968
LSTM	76	0.64989	0.43938	0.00788

Table 4. Comparison results among the proposed models, MLP, CNN and LSTM for USD-to-euro currency exchange rate forecasting.

Network	Parameters	RMSE	MAE	MAPE
QNN	60	0.00281	0.00215	0.00258
HQNN	64	0.00298	0.00224	0.00269
MLP (5,6,3,1)	61	0.00458	0.00352	0.00422
CNN	65	0.00364	0.00274	0.0033
LSTM	76	0.00432	0.00323	0.00388



(a) Mackey-Glass time series.

(b) Lorenz attractor time series for the variable x.

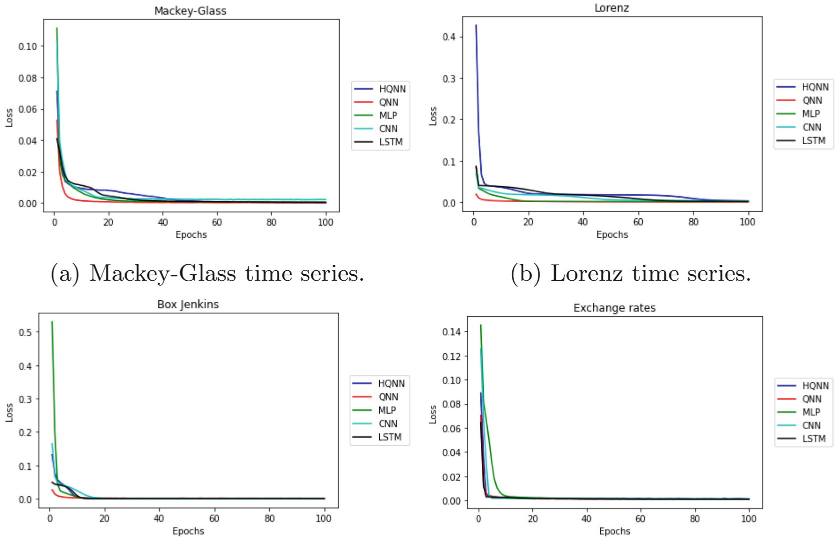
(c) Lorenz attractor time series for the variable y.

(d) Lorenz attractor time series for the variable z.

(e) Box-Jenkins Gas Furnace Data.

(f) Exchange Rate USD/EUR.

Fig. 6. Actual and forecasted outputs using the HQNN.



(a) Mackey-Glass time series.

(b) Lorenz time series.

(c) Box-Jenkins Gas Furnace Data.

(d) Exchange Rate USD/EUR.

Fig. 7. MSE loss curves of the HQNN, QNN, MLP, CNN and LSTM.

6 Conclusions

Two classical-quantum hybrid machine learning architectures with a multilayered structure inspired by the MLP are proposed. In order to evaluate the prediction performance of the models, two univariate time series, Mackey-Glass time series and USD-to-euro currency exchange rate forecasting, as well as two multivariate time series, the Lorenz attractor and prediction of the Box-Jenkins (Gas Furnace) time series are used. The results are competitive compared with those obtained with the MLP, CNN and LSTM with a similar number of trainable parameters, opening the possibility of further exploring how to improve classical models by incorporating quantum computing elements. Additionally, we established a strategy to use quantum architectures in time series forecasting. Future work would focus on exploring more complex hybrid quantum-classical architectures with different embedding circuits and their application to real-world time series data.

Acknowledgment. The authors wish to thank CINVESTAV for the financial support for the realization of this work.

References

1. Abbas, A., Sutter, D., Zoufal, C., Lucchi, A., Figalli, A., Woerner, S.: The power of quantum neural networks. Nat. Comput. Sci. **1**(6), 403–409 (2021)
2. Alam, M., Kundu, S., Topaloglu, R.O., Ghosh, S.: Quantum-classical hybrid machine learning for image classification (ICCAD special session paper). In: 2021 IEEE/ACM International Conference On Computer Aided Design (ICCAD), pp. 1–7. IEEE (2021)
3. Amarasinghe, K., Marino, D.L., Manic, M.: Deep neural networks for energy load forecasting. In: 2017 IEEE 26th International Symposium on Industrial Electronics (ISIE), pp. 1483–1488. IEEE (2017)
4. Barba Maggi, L.M.: Multiscale Forecasting Models. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-94992-5
5. Bausch, J.: Recurrent quantum neural networks. Adv. Neural. Inf. Process. Syst. **33**, 1368–1379 (2020)
6. Bhardwaj, S., Chandrasekhar, E., Padiyar, P., Gadre, V.M.: A comparative study of wavelet-based ANN and classical techniques for geophysical time-series forecasting. Comput. Geosci. **138**, 104461 (2020). https://doi.org/10.1016/j.cageo.2020.104461
7. Biamonte, J.: Universal variational quantum computation. Phys. Rev. A **103**, L030401 (2021). https://doi.org/10.1103/PhysRevA.103.L030401
8. Biamonte, J., Wittek, P., Pancotti, N., Rebentrost, P., Wiebe, N., Lloyd, S.: Quantum machine learning. Nature **549**(7671), 195–202 (2017)
9. Chakraborty, K., Mehrotra, K., Mohan, C.K., Ranka, S.: Forecasting the behavior of multivariate time series using neural networks. Neural Netw. **5**(6), 961–970 (1992)
10. Chandra, R., Goyal, S., Gupta, R.: Evaluation of deep learning models for multistep ahead time series prediction. IEEE Access, 1–1 (2021). https://doi.org/10.1109/ACCESS.2021.3085085
11. Chen, S.Y.C., Yoo, S., Fang, Y.L.L.: Quantum long short-term memory. In: ICASSP 2022–2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 8622–8626. IEEE (2022)
12. Emmanoulopoulos, D., Dimoska, S.: Quantum machine learning in finance: time series forecasting. arXiv preprint arXiv:2202.00599 (2022)
13. Henderson, M., Shakya, S., Pradhan, S., Cook, T.: Quanvolutional neural networks: powering image recognition with quantum circuits. Quantum Mach. Intell. **2**(1), 1–9 (2020). https://doi.org/10.1007/s42484-020-00012-y
14. Hewamalage, H., Bergmeir, C., Bandara, K.: Recurrent neural networks for time series forecasting: current status and future directions. Int. J. Forecast. **37**(1), 388–427 (2021)
15. Hill, T., O'Connor, M., Remus, W.: Neural network models for time series forecasts. Manage. Sci. **42**(7), 1082–1092 (1996)
16. Huang, H.Y., et al.: Power of data in quantum machine learning. Nat. Commun. **12**(1), 1–9 (2021)
17. Hur, T., Kim, L., Park, D.K.: Quantum convolutional neural network for classical data classification. Quant. Mach. Intell. **4**(1), 1–18 (2022)
18. Kolarik, T., Rudorfer, G.: Time series forecasting using neural networks. ACM Sigapl. Apl. Quote Quad **25**(1), 86–94 (1994)
19. Kutvonen, A., Fujii, K., Sagawa, T.: Optimizing a quantum reservoir computer for time series prediction. Sci. Rep. **10**(1), 1–7 (2020)

20. Mari, A., Bromley, T.R., Izaac, J., Schuld, M., Killoran, N.: Transfer learning in hybrid classical-quantum neural networks. Quantum **4**, 340 (2020). https://doi.org/10.22331/q-2020-10-09-340
21. Mitarai, K., Negoro, M., Kitagawa, M., Fujii, K.: Quantum circuit learning. Phys. Rev. A **98**, 032309 (2018). https://doi.org/10.1103/PhysRevA.98.032309
22. Nielsen, M.A., Chuang, I.: Quantum computation and quantum information (2002)
23. Sezer, O.B., Gudelek, M.U., Ozbayoglu, A.M.: Financial time series forecasting with deep learning: a systematic literature review: 2005–2019. Appl. Soft Comput. **90**, 106181 (2020). https://doi.org/10.1016/j.asoc.2020.106181
24. Shi, X., Chen, Z., Wang, H., Yeung, D.Y., Wong, W.K., Woo, W.C.: Convolutional lstm network: a machine learning approach for precipitation nowcasting. In: Advances in Neural Information Processing Systems, vol. 28 (2015)
25. Singh, P.: Fqtsfm: A fuzzy-quantum time series forecasting model. Information Sciences 566, 57–79 (2021). https://doi.org/10.1016/j.ins.2021.02.024, https://www.sciencedirect.com/science/article/pii/S0020025521001663
26. Singh, P., Huang, Y.P.: A new hybrid time series forecasting model based on the neutrosophic set and quantum optimization algorithm. Comput. Ind. **111**, 121–139 (2019). https://doi.org/10.1016/j.compind.2019.06.004, https://www.sciencedirect.com/science/article/pii/S0166361518309072
27. Sridevi, S., Kanimozhi, T., Issac, K., Sudha, M.: Quanvolution neural network to recognize arrhythmia from 2D scaleogram features of ECG signals. In: 2022 International Conference on Innovative Trends in Information Technology (ICITIIT), pp. 1–5. IEEE (2022)
28. Teguri, T., Isokawa, T., Matsui, N., Nishimura, H., Kamiura, N.: Time series prediction by quaternionic qubit neural network. In: 2020 International Joint Conference on Neural Networks (IJCNN), pp. 1–6. IEEE (2020)
29. Ueguchi, T., Matsui, N., Isokawa, T.: Chaotic time series prediction by qubit neural network with complex-valued representation. In: 2016 55th Annual Conference of the Society of Instrument and Control Engineers of Japan (SICE), pp. 1353–1358. IEEE (2016)
30. Wang, H.Z., Li, G.Q., Wang, G.B., Peng, J.C., Jiang, H., Liu, Y.T.: Deep learning based ensemble approach for probabilistic wind power forecasting. Appl. Energy **188**, 56–70 (2017)
31. Zhang, G., Patuwo, B.E., Hu, M.Y.: Forecasting with artificial neural networks: the state of the art. Int. J. Forecast. **14**(1), 35–62 (1998)