Quantum Machine Intelligence
https://doi.org/10.1007/s42484-023-00115-2

RESEARCH ARTICLE

Linear-layer-enhanced quantum long short-term memory for carbon price forecasting

Yuji Cao[1] · Xiyuan Zhou[2] · Xiang Fei[3] · Huan Zhao[2] · Wenxuan Liu[1,2] · Junhua Zhao[1,2]

Received: 4 March 2023 / Accepted: 14 May 2023

Abstract
Accurate carbon price forecasting is important for investors and policymakers to make decisions in the carbon market. With the development of quantum computing in recent years, quantum machine learning has shown great potential in a wide range of areas. This paper proposes a hybrid quantum computing based carbon price forecasting framework using an improved quantum machine learning model. The proposed linear-layer-enhanced Quantum Long Short-Term Memory (L-QLSTM) model employs the linear layers before and after the variational quantum circuits of Quantum Long Short-Term Memory (QLSTM), to extract features, reduce the number of quantum bits and amplify the quantum advantages. The parameter sharing method of the linear layer and the strongly entangled controlled-Z quantum circuit of the variational layer are applied to reduce the parameters and improve the learning performance respectively. We test and evaluate the L-QLSTM based on the practical data of European Union Emission Trading from 2017 to 2020. Results show that the proposed L-QLSTM method can greatly improve the learning accuracy compared to the QLSTM method.

Keywords Carbon price forecasting · Quantum machine learning · Hybrid quantum computing · Quantum long short-term memory · Variational quantum circuit

1 Introduction

With the rise in global temperature and the frequent occurrence of extreme weather, the climate change problem has attracted more attention. Excessive emissions of greenhouse gases are considered one of the most significant causes of climate change (Sun and Huang, 2020). Thus, there has been a pressing need to reduce carbon emissions in the last few decades. Existing methods for carbon emission reduction include using renewable energy resources, establishing carbon emission trading markets, imposing carbon taxes, developing carbon capture technology, arousing the awareness of environmental protection, and so on.

The carbon emission trading market is considered a market-based and relatively economical method to reduce greenhouse gas. European Union Emission Trading Scheme (EU ETS) is the global first major carbon market and has successfully achieved the agreed reduction in carbon emissions (Zhu and Chevallier, 2017). Carbon price prediction is important for the participants in the carbon market. Accurate carbon price forecasting not only helps to establish a robust and stable pricing mechanism, but also provides financial guidance for market investors.

A variety of literature applies traditional statistical models in carbon price forecasting (Niu et al., 2022). Specifically, generalized autoregressive conditional heteroscedasticity (GARCH) and autoregressive integrated moving average (ARIMA) are widely utilized to forecast carbon price (Zhu and Chevallier 2017; Byun and Cho 2013; Zhu et al. 2018). However, these statistical models are based on statistical theory and require the data to meet specific assumptions. This characteristic results in their limited performances in carbon price forecasting owing to the high non-linearity and instability of carbon price time series (Zhu et al., 2019; Huang et al., 2021).

Fuelled by rising computational power, machine learning (ML) approaches have developed into powerful tools for finding complex patterns in data. Among the ML methods, artificial neural networks (ANN) are more potent in describing high non-linearity. Hence, ANN-based meth-

✉ Huan Zhao
paulzhao@cuhk.edu.cn

✉ Junhua Zhao
zhaojunhua@cuhk.edu.cn

Extended author information available on the last page of the article

Published online: 05 July 2023

ods have been applied to various complex problems (Zhao et al., 2020; Liu et al., 2022; Ruan et al., 2022). Given the inherent complex characteristic of carbon price series, ANN-based methods have been preferred recently. Wei et al. (Sun and Zhang, 2018) applied extreme learning machine (ELM) optimized by adaptive whale optimization algorithm to effectively predict carbon price. In reference Fan et al. (2015), the multi-layer perceptron neural network is used for carbon price prediction. Derived from feedforward neural networks, recurrent neural networks can store an internal state/memory and thus better model the temporal data. Long short-term memory (LSTM) is a special kind of recurrent neural network that is well-suited for sequence and temporal dependency data modelling (Hochreiter and Schmidhuber, 1997). Reference Huang et al. (2021) indicates that LSTM can be employed to effectively predict the changes in EU ETS carbon price.

Meanwhile, the recent several years have seen the rise of quantum computing, both in hardware development (Wendin, 2017; Arute et al., 2019) and quantum algorithms (Cong et al., 2019). Quantum computers have the potential to solve problems beyond the reach of classical computers such as factoring large integers (Shor, 1999). Quantum machine learning (QML), an area of the quantum algorithm, is currently receiving a lot of attention. In 2007, Azevedo and Ferreira applied the classical operations and Bloch Sphere representation to mimic the quantum neural networks for stock price forecasting (Azevedo and Ferreira, 2007). However, their work does not involve the actual qubits or qubit operations for computation. In 2018, Mitarai et al. proposed a classical-quantum hybrid QML algorithm using the variational quantum circuit (VQC) (Mitarai et al., 2018). The VQC can learn a given task by tuning parameters implemented on it. Compared to the classical computational model, VQCs have more expressive power by utilizing the characteristics of quantum entanglement between quantum bits (qubits) and are able to address the time series forecasting tasks. Ceschini appended a VQC after stacked LSTM to forecast renewable energy time series (Ceschini et al., 2022). Chen proposed a quantum LSTM (QLSTM) by replacing neural networks with VQCs, and the experiment results show that QLSTM learns faster and converges more stably than classical LSTM in certain problems (Chen et al., 2022).

To take advantage of LSTM in temporal learning and utilize the expressive power granted by quantum entanglement, this paper proposes a novel hybrid quantum computing framework for carbon price forecasting with the linear-layer-enhanced Quantum LSTM (L-QLSTM) model. The contributions are summarized as follows:

- A specifically designed hybrid quantum computing framework with a quantum machine learning model is proposed for carbon price forecasting. To the best of our knowledge, this paper is among the first to apply hybrid quantum computing methods to carbon price forecasting.
- The L-QLSTM model is proposed, which employs linear layers before and after the VQC to extract feature representations. By compressing input features, linear layers reduce the number of qubits and considerably increase the learning ability of VQCs.
- The parameter sharing method is applied for the linear layers before the VQCs to reduce parameters without much impact on the prediction accuracy.
- The variational layer in the VQCs is replaced by a strongly entangled controlled-Z quantum circuit. Quantum gates and connections between different quantum bits are optimized for stronger quantum entanglement and better expressibility.

The remaining sections are organized as follows. Section 2 briefly presents the background of VQC and QLSTM. Then the proposed framework and linear-layer-enhanced QLSTM model are introduced in Sect. 3. Section 4 shows the experiment results of the proposed method with the EU ETS data. Conclusion and further discussion are provided in the last section.

2 Background

2.1 Variational quantum circuit

A quantum circuit is a model for quantum computation that consists of a number of qubits (quantum bit, a basic unit of quantum information). The computation process is achieved by changing the states of qubits with a sequence of quantum gates. The result of computation is then measured at the end of the circuit.

Variational quantum circuit (VQC) is a special category of quantum circuits with learnable parameters that can be optimized iteratively. A general VQC architecture is shown in Fig. 1, which has three major components:

1. Encoding Layer: Encode the classical data into quantum state by changing the states of qubits from reference state

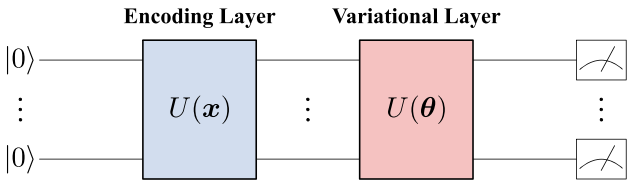

Fig. 1 A general architecture for variational quantum circuits. $U(\boldsymbol{x})$ represents quantum operations for encoding classical input data $\boldsymbol{x}$ and $U(\boldsymbol{\theta})$ represents the variational circuit part with learnable parameters $\boldsymbol{\theta}$

$||0\rangle$ to the target states with operation $U(\boldsymbol{x})$. Typically, each qubit encodes one classical inputs features.
2. Variational Layer: Entangle and rotate the qubits to the target state with operation $U(\boldsymbol{\theta})$. Variational layer is the actual learnable part of a VQC. The quantum gates are attached with tunable parameters.
3. Measurement Layer: Output the computational result of the VQC by measuring the quantum states of the qubits. Measurement of one qubit transforms its quantum state into classical data.

Note that the number of qubits and the number of measurements are free to be adjusted based on the problem. Similarly, the number of green dashed boxes in the variational layer can be tuned based on the expressive power (model complexity) of the circuit.

2.2 Quantum long short-term memory

Quantum LSTM (QLSTM) model is the quantum version of the classical LSTM model. Compared with the classical LSTM model, QLSTM replaces the classical neural networks in different gates with VQCs. By substituting the neural networks with VQCs, QLSTM learns faster and converges more stably (Chen et al., 2022). The structure is shown in Fig. 2 and the corresponding forward pass equations in a compact form are as follows:

$$v_t = [h_{t-1}, x_t] \tag{1}$$
$$f_t = \sigma(VQC_1(v_t)) \tag{2}$$
$$i_t = \sigma(VQC_2(v_t)) \tag{3}$$
$$\tilde{c}_t = \tanh(VQC_3(v_t)) \tag{4}$$
$$c_t = f_t \otimes c_{t-1} + i_i \otimes \tilde{c}_t \tag{5}$$
$$o_t = \sigma(VQC_4(v_t)) \tag{6}$$
$$h_t = VQC_5(o_t \otimes \tanh(c_t)) \tag{7}$$
$$y_t = VQC_6(h_t) \tag{8}$$

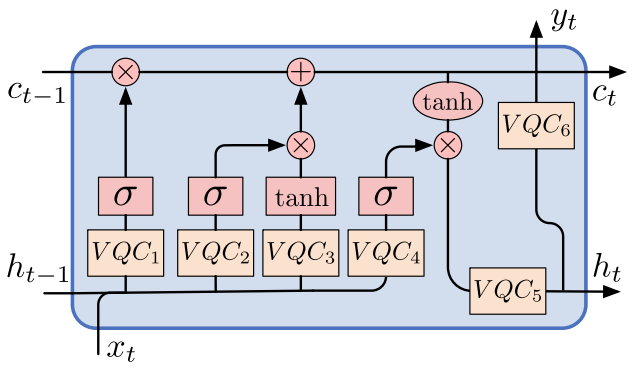

Fig. 2 Quantum LSTM model. $\oplus$ and $\otimes$ represent element-wise plus and product, respectively

where the subscript t indexes the time step; x and h stand for the input data and hidden state, respectively; $[\cdot]$ represents the vector concatenation; f, i, o represent the forget gate, input gate, and output gate in the LSTM model, respectively; c_t is the cell state; $\tilde{c}_t$ is cell input vector; $\sigma(\cdot)$ and tanh represent the activation function of Sigmoid and hyperbolic tangent, respectively; $\otimes$ denotes the Hadamard product (element-wise product) and y_t represents the output.

3 Framework

Carbon price is typical time series data and the prediction accuracy is closely related to the non-linear feature in temporal. For example, in the late Phase III of EU ETS, the carbon price dramatically climbed up to €35 from €5 and fluctuated considerably (Huang et al., 2021), making the forecasting work more challenging. With the development of quantum machine learning, more and more areas are trying to take quantum advantages and apply quantum machine learning methods to traditional problems.

In this paper, we propose a hybrid quantum computing framework that employs a linear-layer-enhanced quantum long short-term memory (L-QLSTM) model for carbon price forecasting. Compared with other classical frameworks, the proposed framework distributes the learning tasks to both quantum computers and classical computers based on their advantages. Quantum computers are responsible for computing and classical computers are responsible for features mapping and updating model parameters. Also, a data engineering module is designed as an auxiliary module to provide processed data for L-QLSTM model learning. After model learning, an online model serving stage is proposed for predicting carbon price in real time.

The overall framework is illustrated in Fig. 3, which includes an offline learning stage and an online real-time forecasting stage. At the offline stage, the data engineering module collects features and carbon price from EU ETS dataset. The collected data is then normalized and transformed for learning. Afterward, the processed data is stored in a database that prepares the required dataset for L-QLSTM model training. Before the model learning part, classical neural networks and VQCs are initialized. After initialization, the dataset is passed into a hybrid computation model. In the hybrid computation model, the classical neural networks first squeeze the input data as a compressed feature representation for following VQCs. The VQCs then perform the required computation by rotating and entangling the qubits. The computation result of the VQCs is measured and passed to classical neural networks to obtain the predicted price. To minimize the forecasting error, the framework calculates the loss between the predicted price and the true price. According to the loss, an optimizer updates both the learn-

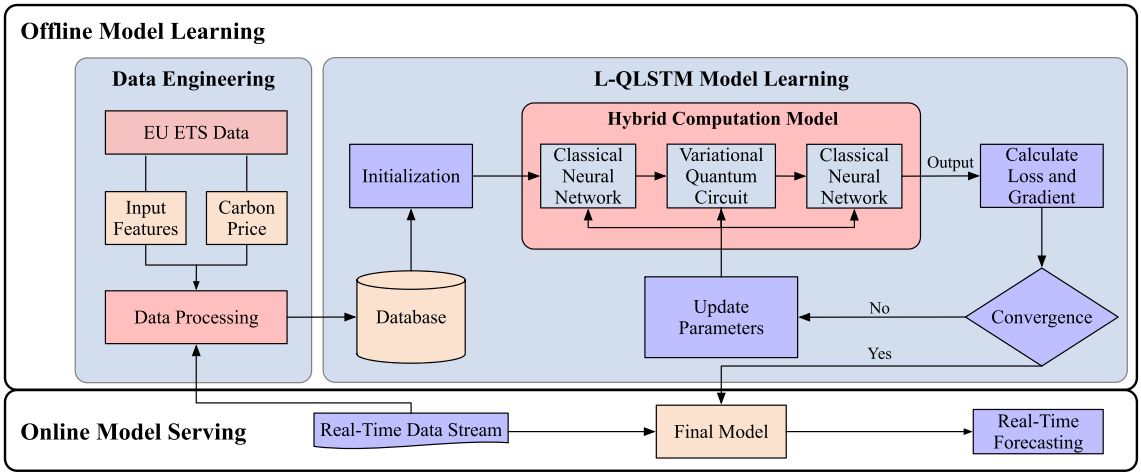

Fig. 3 Hybrid quantum computing based carbon price forecasting framework

able parameters in classical neural networks and VQCs by backpropagating the loss function until convergence. After the model is well trained, it can be applied online for real-time forecasting. In the online stage, a real-time data stream is passed to the final model for real-time forecasting. Along the process, the data stream is processed and stored in the database for further learning.

3.1 Linear-layer-enhanced quantum long short-term memory model

QLSTM performs well in time series forecasting tasks with relatively regular features. However, there are some limitations with the application of QLSTM. A graphical interpretation is shown in Fig. 4. In the VQC, encoding and decoding employ a one-to-one way. For a timestep, suppose that the hidden state $\boldsymbol{h}$ has m features and input features $\boldsymbol{x}$ has n features, the VQC should use $m+n$ qubits to encode classical features. However, since the dimension of the VQC's output should match the dimension of the hidden state $\boldsymbol{h}$, the number of qubits to be measured should be m and quantum information of the remaining n qubits is lost and wasted during the learning process. Also, redundant features cost more qubits usage and may strongly impede the learning ability of QLSTM, making it infeasible in the application.

The above limitations can be addressed with the embedding layer. Embedding layer is a classical neural network layer in deep learning that maps the input features into a target dimension to obtain the feature representations in the target dimension. Linear layer is a simple but effective embedding layer, called linear embedding layer, to compress features. Specifically, linear embedding layers use matrix multiplication to transform the input features from the input dimension to the output dimension. When the target dimension is smaller than the input dimension, it outputs in a compressed feature representation. Based on this idea, a linear-layer-enhanced QLSTM model shown in Fig. 5 is proposed to address the limitations. Based on QLSTM, the proposed model inserts a shared linear embedding layer before the VQCs and four separate linear layers after each VQC. The shared linear embedding layer serves as a feature compressor for the following VQCs, allowing all qubits to be measured and avoiding wasting quantum information. The four separate linear layers remap the output of VQCs and enable the qubits to be adjusted freely.

The process of a forward pass is as follows. First, the hidden state from the last timestep h_{t-1} and input features of current timestep x_t are concatenated as v_t. Then the v_t is passed to a linear embedding layer for compressing. After

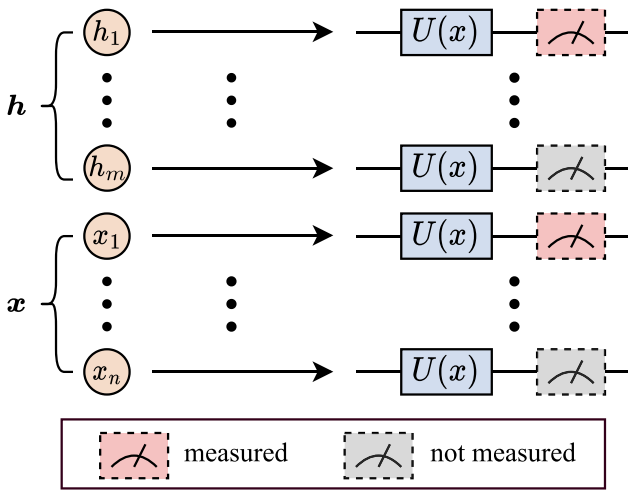

Fig. 4 One-to-one way is employed in the encoding and decoding part of the VQC. m out of $m+n$ qubits are measured and quantum information of the remaining n qubits is wasted, where the set of m measurements is selected from $m+n$ qubits and fixed along the process

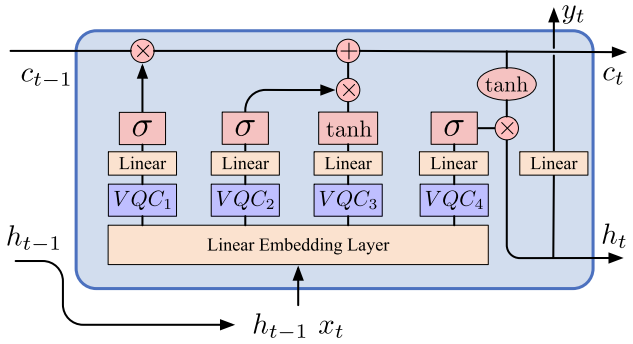

Fig. 5 Linear-layer-enhanced QLSTM model

that, the compressed features are passed to four different VQCs for computation. The four computation results are remapped to the hidden state dimension by four linear layers, respectively. At last, non-linearity is introduced by different activation functions. The results are connected and processed with different operations to achieve the characteristics of temporal learning with long dependency. The formulas in a forward pass are as follows:

$$z_t = L_e(v_t) \tag{9}$$
$$f_t = \sigma(L_1(VQC_1(z_t))) \tag{10}$$
$$i_t = \sigma(L_2(VQC_2(z_t))) \tag{11}$$
$$\tilde{c}_t = \tanh(L_3(VQC_3(z_t))) \tag{12}$$
$$c_t = f_t \otimes c_{t-1} + i_i \otimes \tilde{c}_t \tag{13}$$
$$o_t = \sigma(L_4(VQC_4(z_t))) \tag{14}$$
$$h_t = o_t \otimes \tanh(c_t) \tag{15}$$
$$y_t = L_o(h_t) \tag{16}$$

where L_e represents the linear embedding layer; z_t represents the compressed representation of v_t; L_i stands for the linear layer after VQC_i, and L_o represents the linear layer in the end to obtain the forecasting value.

3.2 Parameter sharing linear embedding layer

Linear embedding layer maps features to a target dimension. In the L-QLSTM model, the linear embedding layer before a VQC compresses input features for the VQC. A common way of embedding is to provide separate feature embeddings for separate VQCs, which is shown in Fig. 6. For a VQC i, the separate linear embedding layer maps the input data $[h_{t-1}, x_t]$ to a compressed feature representation $z_{t,i}$. By providing different mappings, the separate linear embedding layers can fit the non-linearity well. However, a large number of parameters are required and the relation between different VQCs is ignored.

In the proposed model, a shared linear embedding layer is applied for different VQCs, as shown in Fig. 6. It maps the input data to a shared compressed feature representation z_t. In the learning process, the shared feature representation is optimized for four different VQCs. Therefore, the dependency and relations between different VQCs are involved in the optimization, which increases the learning ability. Moreover, the parameter sharing method reduces the overfitting risks. Generally, when the number of parameters is larger than required, the model starts to fit the random noises in the training dataset. Under such circumstance, the parameter sharing method reduces the model parameters and thus avoids the overfitting
problem.

The linear embedding layer after the VQC maps the output to the dimension of the hidden state h_t. For the linear embedding layers after each VQC, the separate way is preferred. Since the VQCs in L-QLSTM are similar to the gates in LSTM, different VQCs are responsible for different functions. A shared linear embedding layer cannot extract different information according to the functions of each VQC. Therefore, the separate linear embedding layers are more suitable when inserted after the VQCs.

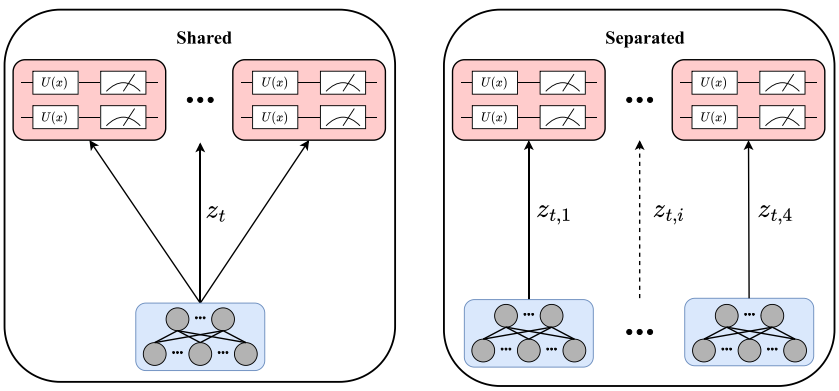

Fig. 6 Two types of linear embedding layer

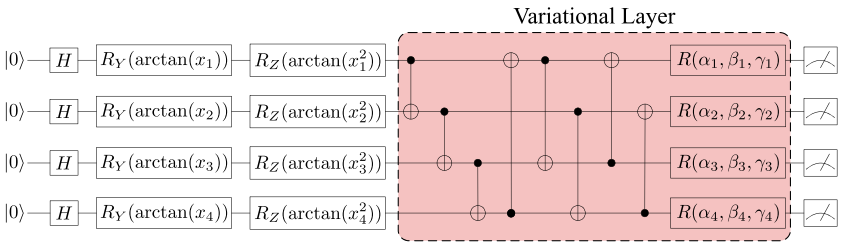

Fig. 7 Variation quantum circuits in the QLSTM model. H, R represent quantum gate and x is the vector of classical input data. The parameters (α, β, γ) in the variational layer are learnable. The line connected with $\bullet$ and $\oplus$ represents controlled-NOT (CNOT) gate

3.3 Strongly entangled controlled-Z variational layer

Variational layer is the key part of VQC to accomplish the learning accurately. In the variational layer, qubits are entangled and rotated from the reference state to the target state, which enables non-linear complex mapping of information. Therefore, the mapping properties of variational layer can significantly impact the prediction accuracy of a variational quantum model.

The variation quantum circuit of the QLSTM model is shown in Fig. 7, and quantum entanglement is achieved by CNOT gates. However, since CNOT gates are not completely connected for all qubits pairs, the entanglement between qubits is not strong enough. Thereby, the lack of qubit connections impedes the learning ability of the circuit. Also, the position of variational rotations is a factor affecting the performance. Since the variational rotations $R(\alpha_i, \beta_i, \gamma_i)$ are performed in the end, qubits cannot fully utilize the quantum entanglement for rotation and thus cannot well represent the solution space.

To optimize the performance of the proposed model, the variational layer in the QLSTM's VQC is substituted by a strongly entangled controlled-Z (SCZ) variational layer (Fig. 8). The strongly entangled circuit connects all qubit pair combinations for strong quantum entanglement and controlled-Z gates rotate the qubit along the Z-axis for computation. With the applied variational layer, the proposed improves the original VQC in the following ways. First, CNOT gates are replaced with controlled-Z rotation gates and thus variational rotations are performed during the quantum entanglement process. Therefore, the qubits can take advantage of the quantum entanglement for rotation. Moreover, compared with CNOT gates, the controlled-Z rotation gates are likely to feature in many experiments since they are inherently implemented in superconducting architectures (Sim et al., 2019). Second, the proposed VQC comprises all-to-all connectivity between all qubit pairs. The increasing connectivity leads to both favorable expressibility and entangling capacity (Sim et al., 2019).

3.4 Parameters learning

Similar to classical neural networks, the parameters of VQCs can be optimized by the gradient-based approach. Given an output $f(x, \theta)$ with respect to quantum parameters θ and input features x, the gradient of the VQC can be calculated by the parameter-shift rule as below:

$$\nabla_\theta f(x, \theta) = \frac{1}{2}\left[f(x, \theta + \frac{\pi}{2}) - f(x, \theta - \frac{\pi}{2})\right] \tag{17}$$

Thereby, the loss function L can be minimized by backpropagating the gradients between VQCs and classical neural networks, and the whole hybrid quantum computing model is optimized iteratively.

The algorithm for the learning process of L-QLSTM is shown in Algorithm 1. For each epoch, a batch of x and y is extracted from the training set and passed into the L-QLSTM model. Then based on output $\hat{y}$ and true value y, the loss is calculated. After that, the gradient is set to 0 and the learnable parameters θ are updated by backpropagating the loss L. The training process is looped until the model is converged.

4 Case study

4.1 Settings

4.1.1 Data sets

As the biggest carbon trading market in the EU ETS, the European Climate Exchange (ECX) is an indicator of the global carbon markets. This paper forecasts the settlement prices of the EUA continuous futures contracts that are traded

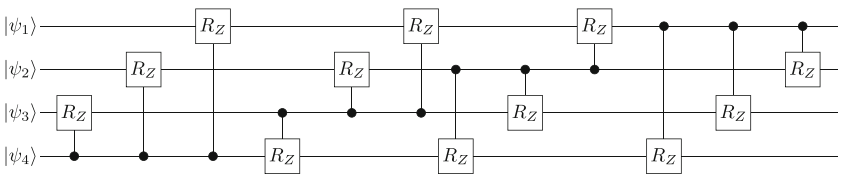

Fig. 8 Strongly entangled controlled-Z (SCZ) variational layer

Algorithm 1 L-QLSTM Model Learning Process.

Input: Training set D_{train}; number of training epochs N; convergence threshold δ
Output: Well trained L-QLSTM model
1: *Initialization*: Initialize the parameters in VQCs and linear layers θ; initialize the hidden state h_0 to 0; initialize loss L to $+\infty$
2: **while** number of epochs $< N$ and $L \geq \delta$ **do**
3: **for** (x, y) in D_{train} **do**
4: Calculate predicted carbon price $\hat{y}$ by feeding x into the L-QLSTM model;
5: Calculate loss L between y and $\hat{y}$;
6: Set all gradients to 0;
7: Update learnable parameters θ in the VQCs and linear layers by backpropagating the loss L;
8: **end for**
9: **end while**

in the ECX. The daily data is collected from the European Energy Exchange in Euros/ton. The raw data includes the date, the opening price, the closing price, the highest price, the lowest price, and the trading volume of each trading day. Experiments use carbon prices from January 1, 2017, to December 31, 2020 (excluding public holidays). The data is split into three datasets without shuffling: training set (0% $\sim$ 64%), validation set (64% $\sim$ 80%) and test set (80% $\sim$ 100%). Figure 9 shows the carbon prices of the chosen period and the corresponding dataset division.

After collection and division, the dataset is further pre-processed for model learning and testing. First, both input features and carbon price are min-max normalized into [0, 1] for numerical stability and faster convergence. Then the data of D_{t-1}, D_{t-2} and D_{t-7} is selected for the carbon price forecasting of the trading day D_t, where the holidays are deleted for consistency. For each trading day, the one volume and 4 price-related features are used as input. Thereby, the data sample is of shape 3×5.

In the training process, Mean Squared Error (MSE) is used as the training loss and the Adam optimizer is used for updating both the parameters of neural networks and VQCs.

Fig. 9 Selected carbon price in the late Phase III of EU carbon market and corresponding dataset division

4.1.2 Evaluation metrics

Models are evaluated by Root Mean Square Error (RMSE) and Mean Absolute Error (MAE). Computing the RMSE corresponds to the L2 norm (i.e., the Euclidean norm); computing MAE corresponds to the L1 norm (i.e., the Manhattan norm). In particular, as a rule of thumb, RMSE is more sensitive to outliers than MAE (Chicco et al., 2021). The formulas are as follows:

$$RMSE = \sqrt{\frac{1}{N}\sum_{t=1}^{N}(y_t - \hat{y}_t)^2}$$

$$MAE = \frac{1}{N}\sum_{t=1}^{N}\left\|y_t - \hat{y}_t\right\|$$

where y_t is the normalized true value and $\hat{y}_t$ is the normalized predicted value for t^{th} day.

Note that all models are selected by minimum validation loss. Additionally, to avoid randomness, we randomly picked 10 seeds and repeated the experiments of each model over them. The average performance is calculated as the result.

4.1.3 Hyperparameter and implementation details

The hyperparameters for the methods are shown in Table. 1. The neural network-involved model is with the batch size of 1, the learning rate of 0.001, depth of 1, hidden dimension of 4, input dimensions of 3×5 (see Sect. 4.1.1) and output dimensions of 1. All the neural network-involved models are trained over a minimum of 300 epochs and a maximum of 600 epochs with an early stopping technique to prevent overfitting (Ying, 2019). The loss of the validation set is measured every 10 epochs, and the learning is early stopped if the loss decreasing for 5 consecutive times is less than 10^{-6}.

The qubits number of L-QLSTM is set to 4. Since the hidden state dimension is 4 and the qubits number of QLSTM should be equal to the sum of the input dimension and the hidden state dimension, the qubits number of QLSTM is set to 9.

The algorithms are implemented with PyTorch (Paszke et al., 2019) and PennyLane (Bergholm et al., 2018). The VQC is simulated on the lightning.qubit device in the Pen-

Table 1 Test parameters

Parameter	Value	Parameter	Value
Input Dimension	3×5	Depth	1
Output Dimension	1	Hidden Dimension	4
Batch Size	1	Maximum Epoch	600
Learning Rate	0.001	Minimum Epoch	300

nyLane. To foster reproducibility and further exploration of our work, we have established a publicly accessible GitHub repository that contains the complete implementation of our model, including the source code, relevant data, and comprehensive documentation. The repository can be accessed in https://github.com/TravisCao/L-QLSTM.

The dimension of the hidden state determines the temporal modelling capacity. For the fairness concern, we set the hidden state dimension to be 4 for LSTM, QLSTM and L-QLSTM. The numbers of model parameters for LSTM, QLSTM and L-QLSTM are 181, 162 and 205, including both neural network parameters and VQC parameters. For LSTM, 181 represents the 176 parameters of the LSTM layer (three gates and one cell state that act on a 4×5 weight matrix for the input layer, a 4×4 weight matrix for the hidden layer, a bias vector of size 4×1 for the input layer, and a bias vector of size 4×1 for the hidden layer) and 5 parameters for the linear layer (a bias vector of size 4×1 for the output layer and output dimension 1×1). For QLSTM, 162 represents 6 VQCs with each of 27 parameters (9 qubits with each of 2 encoding parameters and 1 rotational parameter). For L-QLSTM, 205 stands for 4 VQCs with each of 20 parameters (in total 80 parameters) and 6 linear layers (in total 125 parameters).

4.2 Performance comparisons of L-QLSTM, QLSTM and LSTM

To verify the improvement of L-QLSTM over QLSTM and compare the difference between quantum methods and the classic methods, the performance of LSTM, QLSTM and L-QLSTM are compared. Figure 10 shows the comparison of prediction performance. As shown in the graph, L-QLSTM variants can predict temporal changes well, and its prediction performance is comparable to LSTM, while QLSTM cannot learn.

Fig. 10 Comparison of prediction results between LSTM, QLSTM and L-QLSTM

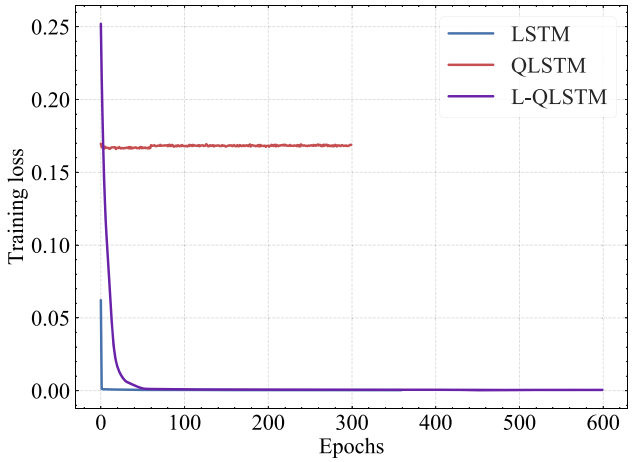

Fig. 11 Comparison of losses between LSTM, QLSTM, and L-QLSTM

Figure 11 demonstrates the training losses (MSE) between LSTM, QLSTM and L-QLSTM, respectively. From the figure, it can be seen that the training loss of QLSTM remains flat at the initial level. Therefore, QLSTM stops training after 300 epochs since we set the early stopping technique with a minimum epoch of 300.

The underlying reason for the failure of the QLSTM approach is the information loss during measurement. In the experiment, the dimension of input features is 5, and the dimension of hidden states is 4. For QLSTM, due to the one-to-one encoding constraint, 9 qubits should be used according to the sum of input features and hidden states. However, in the measurement process, only 4 out of 9 (44%) qubits can be used for the next block as hidden states, while the other 5 qubits are ignored. Therefore, most information is lost after measurement and thereby information obtained by the measurement has a limited expressibility for the complex carbon price time series. Since the quantum measurements are used to retrieve the values for optimizing the VQCs, limited information in the measurement makes the model fail to compute correct gradients for updating the parameters and minimizing the training loss. Therefore, the training loss of QLSTM remains flat at the initial level during the training process. Compared to QLSTM, L-QLSTM employs the linear embedding layer to compress both the input features and the hidden states. Thereby, L-QLSTM squeezes the dimension of 9 to 4 for the following VQCs, which allows all the qubits in L-QLSTM to be measured, resulting in an improvement of L-QLSTM over QLSTM.

Table 2 shows the prediction errors of LSTM, QLSTM and L-QLSTM on training and test sets. The proposed L-QLSTM model significantly outperforms the fully quantum QLSTM model, as evidenced by lower RMSE and MAE values for both training and test sets. Although the L-QLSTM model's error rates are slightly higher than those of the classical LSTM model, they remain competitive, demonstrat-

Table 2 Prediction errors between LSTM, QLSTM and L-QLSTM

Prediction Error		Model QLSTM	LSTM	L-QLSTM
Training Set	RMSE	0.2863	0.0188	0.0256
	MAE	0.2563	0.0131	0.0170
Test Set	RMSE	0.3024	0.0297	0.0436
	MAE	0.2723	0.0223	0.0322

ing that the L-QLSTM model effectively bridges the gap between quantum LSTM and classical LSTM. This suggests that incorporating linear layers for compressing the features before the VQC, as in the L-QLSTM, could be a promising avenue for future carbon price prediction research.

4.3 Comparative analysis the effect of linear embedding layer

To illustrate the effect of the linear embedding layer in the proposed model, we compare QLSTM and L-QLSTMs with different linear embedding layer types. As stated in Sect. 3.1, due to the existence of the linear embedding layer, the number of qubits can be arbitrarily selected and more qubit information can be measured, thereby improving the learning ability of the model. In this experiment, we chose the same VQC structure for QLSTM and L-QLSTM for consistency. For QLSTM, the number of model parameters is 162. For L-QLSTM, there are different combinations in the selection of linear layers before and after VQCs. For the reason of readability, we use (before, after) as the representation of the combination. The number of parameters of (shared, separate), (separate, separate) and (shared, shared) are 173, 293 and 113, respectively.

Table 3 shows the prediction errors between QLSTM and L-QLSTMs with different linear embedding layer combinations. For L-QLSTMs, L-QLSTM with (shared, separate) linear embedding layers performs best. Compared with the model of (separate, separate) linear embedding layer type, the difference between the two models' errors is in the thousandths. As stated in Sect. 4.1.2, the RMSE of the L-QLSTM that shares a linear embedding layer before VQCs is smaller, indicating that the predicted value has fewer outliers. Considering that the number of parameters is much larger when using four separate linear embedding layers before VQCs than that of sharing one linear embedding layer, it can be concluded that L-QLSTM sharing one linear embedding layer before VQCs can reduce the parameters. Compared with the model with (shared, shared) linear embedding layers combination, the L-QLSTM with four separate linear embedding layers after the VQCs has a better prediction performance. Combined with Sect. 3.2, the following explanation is given. Before VQCs, a shared linear embedding layer can optimize dependency and relationship between different VQCs. Therefore, with a smaller number of parameters, similar prediction performance is achieved and the occurrence of outliers is reduced. However, after VQCs, since VQCs are similar to the gates in classical LSTM, different VQCs are responsible to different functions and output different information. Separate linear embedding layers can extract the output information of different VQCs, while shared linear embedding layer may confuse the information. Therefore, the separate linear embedding layers are more suitable when inserted after the VQCs.

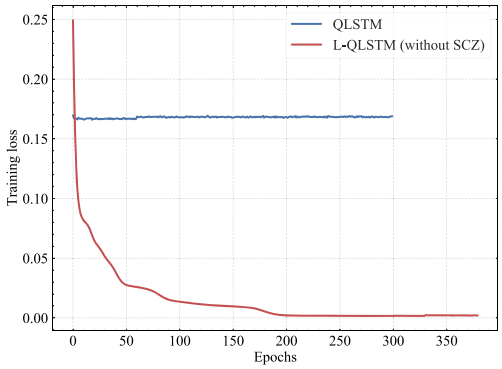

Fig. 12 Comparison of losses between QLSTM and L-QLSTM

Table 3 shows that L-QLSTM has a great improvement over QLSTM. On the training set, the RMSE predicted by L-QLSTM is 85.61% smaller than that predicted by QLSTM, and the MAE is 88.29% smaller. On the test set, the RMSE predicted by L-QLSTM is 77.88% smaller than that predicted by QLSTM, and the MAE is 82.04% smaller. Therefore, we conclude that L-QLSTM has a greater improvement in prediction accuracy than QLSTM.

Table 3 Prediction errors for QLSTM and different L-QLSTMs

	Linear Embedding Layer		Training Set		Test Set	
	Before VQC	After VQC	RMSE	MAE	RMSE	MAE
QLSTM	/	/	0.2863	0.2563	0.3024	0.2723
	Shared	**Separate**	**0.0412**	0.0300	**0.0669**	0.0489
	Separate	Separate	0.0438	**0.0283**	0.0687	**0.0456**
	Shared	Shared	0.0574	0.0389	0.0703	0.0510
Error Reduction	/	/	85.61%	88.29%	77.88%	82.04%

Table 4 Prediction errors between L-QLSTM (Original) and L-QLSTM (SCZ)

Prediction Error		Model Original	SCZ	Error Reduction
Training Set	RMSE	0.0412	**0.0256**	37.86%
	MAE	0.0300	**0.0170**	43.33%
Test Set	RMSE	0.0669	**0.0436**	34.83%
	MAE	0.0489	**0.0322**	34.15%

Figure 12 shows the comparison of losses between QLSTM and L-QLSTM. It is clear that QLSTM has almost no predictive ability and cannot converge, while L-QLSTM predicted values are similar to real values and converge well. The reason behind the phenomenon is that QLSTM cannot measure all qubits due to the lack of linear embedding layers, making some qubits unmeasured and thus wasting the quantum information.

4.4 Comparison of different variational layers

To verify the improvement of strongly entangled controlled-Z variational layer prediction ability, we compared the L-QLSTM with different variational layers, including the original variational layer using the CNOT gate and rotation gate (VQC in Fig. 7), and the proposed strongly entangled controlled-Z variational layer (VQC in Fig. 8). The number of parameters for L-QLSTM with the original variational layer and SCZ is 173 and 205, respectively.

Table 4 shows the prediction errors for different variational layers. On the training set, the RMSE predicted by L-QLSTM with SCZ is 37.86% smaller than that predicted by L-QLSTM with the original variational layer, and the MAE is 43.33% smaller. On the test set, the RMSE predicted by L-QLSTM with SCZ is 34.83% smaller than that predicted by L-QLSTM with the original variational layer, and the MAE is 34.15% smaller. We found that L-QLSTM with SCZ has improved prediction accuracy more than the original variational layer.

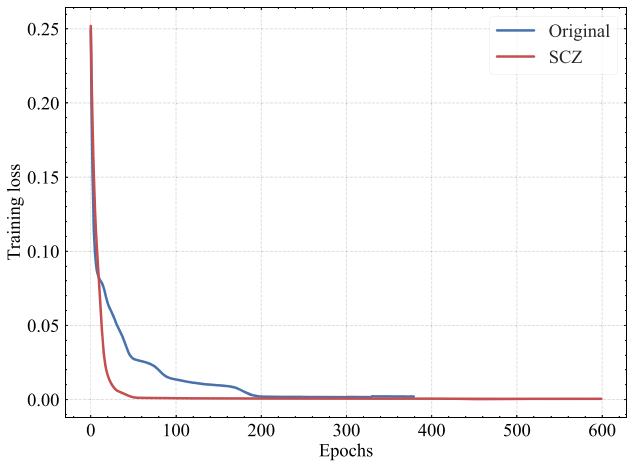

Fig. 13 Loss Comparison of L-QLSTM (Original) and L-QLSTM (SCZ)

Table 5 Prediction errors for different hidden dimension models

Prediction Error		Hidden Dimension 2	4	6
Training Set	RMSE	**0.0249**	0.0256	0.0257
	MAE	**0.0167**	0.0170	0.0167
Test Set	RMSE	**0.0411**	0.0436	0.0417
	MAE	**0.0306**	0.0322	0.0311

Figure 13 compares losses between L-QLSTM with the original variational layer and L-QLSTM with SCZ. L-QLSTM with SCZ has a more significant improvement in the convergence speed and stability of loss than the original with SCZ. The experimental result shows that SCZ variational layer can improve the performance of L-QLSTM. Two reasons are concluded for the improvement of the SCZ: (1) the CNOT gates are replaced with controlled-Z rotation gates, which improves the data transmission efficiency in the network topology (Dg et al., 2018); (2) SCZ completely connects all qubits pairs, resulting a strong quantum entanglement and thus increasing the learning ability.

4.5 Parameter sensitivity analysis

Generally, the results of a robust L-QLSTM should not change much if the hidden dimension changes within a reasonable range. To effectively prove that the prediction results of the L-QLSTM are universal, we compared L-QLSTM with hidden layer dimensions of 2, 4 and 6, with parameters of 155, 205, and 255, respectively.

Table 5 shows the prediction errors for different models. The results show that difference between the two errors of the training set and the test set is in the thousandths, most of the error changes are less than 5%. Therefore, the phenomenon indicates that within the range of the chosen hidden dimension, the model performance is not sensitive to the selection of hidden dimension. The reason may be linked to the existence of VQC. By replacing the classical neural networks with VQCs in the L-QLSTM model, the process of information transfer is more efficient and more stable, making L-QLSTM being insensitive to the hidden dimension.

5 Conclusion

This paper proposes a hybrid quantum computing framework based on the QLSTM model for carbon price forecasting problems. The L-QLSTM model is based on QLSTM and lin-

ear layers to improve the quantum model learning ability. In particular, linear layers are applied before and after the VQCs to extract feature representations, reduce quantum bit usage, and enhance the expressive advantage in quantum circuits. The parameter-shared method is utilized for the linear layers before VQCs to reduce the parameters. Moreover, an optimized VQC is applied with stronger quantum entanglement that improves the learning ability of the quantum circuit. The test results show that (1) compared to the classical LSTM, the proposed framework can achieve a comparable accuracy; (2) the linear embedding layers significantly improve the quantum model performance by compressing the features and allowing all qubits to be measured; (3) the linear layer parameter sharing method before VQCs can reduce the number of parameters without much impact on the prediction results; and (4) the optimized variational quantum circuit further improves the model performance with stronger quantum entanglement and rotation operations. The work is among the first to successfully predict the carbon price with a hybrid quantum computing method. Further work will focus on the possibility of other quantum machine learning methods in carbon price prediction problems.

Author contribution Yuji Cao wrote the manuscript, designed and implemented the proposed framework, and conducted the experiments. Xiyuan Zhou contributed to the manuscript by writing the case study section and testing different model settings. Xiang Fei contributed to the design of the framework. Huan Zhao, Wenxuan Liu, and Junhua Zhao provided critical guidance and supervision throughout the research process, and contributed to the refinement of the manuscript. All authors reviewed and approved the final version of the manuscript.

Funding This research work is supported by the National Natural Science Foundation of China, Grant/Award Numbers: 71931003, 72061147004, 72171206, 72192805, 42105145; Guangdong Province Natural Science Foundation (No. 2023A1515011438); Shenzhen Institute of Artificial Intelligence and Robotics for Society.

Data availability All of the material and data is owned by the authors and/or no permissions are required.

Declarations

Conflict of interest The authors declare no competing interests.

References

Arute F, Arya K, Babbush R et al (2019) Quantum supremacy using a programmable superconducting processor. Nature 574(7779):505–510. https://doi.org/10.1038/s41586-019-1666-5

Azevedo CR, Ferreira TA (2007) Time series forecasting with qubit neural networks. In: Conf. ASC'07 Proc. Elev. IASTED Int

Bergholm V, Izaac J, Schuld M, et al (2018) Pennylane: Automatic differentiation of hybrid quantum-classical computations. ArXiv Prepr ArXiv181104968 https://arxiv.org/abs/1811.04968

Byun SJ, Cho H (2013) Forecasting carbon futures volatility using GARCH models with energy volatilities. Energy Economics 40:207–221. https://doi.org/10.1016/j.eneco.2013.06.017

Ceschini A, Rosato A, Panella M (2022) Hybrid quantum-classical recurrent neural networks for time series prediction. In: 2022 Int. Jt. Conf. Neural Netw. IJCNN. IEEE, pp 1–8

Chen SYC, Yoo S, Fang YLL (2022) Quantum long short-term memory. In: ICASSP 2022-2022 IEEE Int. Conf. Acoust. Speech Signal Process. ICASSP. IEEE, pp 8622–8626

Chicco D, Warrens MJ, Jurman G (2021) The coefficient of determination R-squared is more informative than SMAPE, MAE, MAPE, MSE and RMSE in regression analysis evaluation. PeerJ Comput Sci 7:e623. https://doi.org/10.7717/peerj-cs.623

Cong I, Choi S, Lukin MD (2019) Quantum convolutional neural networks. Nat Phys 15(12):1273–1278. https://doi.org/10.1038/s41567-019-0648-8

Dg Zhang, Zhang T, Dong Y et al (2018) Novel optimized link state routing protocol based on quantum genetic strategy for mobile learning. Journal of Network and Computer Applications 122:37–49. https://doi.org/10.1016/j.jnca.2018.07.018

Fan X, Li S, Tian L (2015) Chaotic characteristic identification for carbon price and an multi-layer perceptron network prediction model. Expert Systems with Applications 42(8):3945–3952. https://doi.org/10.1016/j.eswa.2014.12.047

Hochreiter S, Schmidhuber J (1997) Long short-term memory. Neural Comput 9(8):1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735

Huang Y, Dai X, Wang Q et al (2021) A hybrid model for carbon price forecastingusing GARCH and long short-term memory network. Applied Energy 285(116):485. https://doi.org/10.1016/j.apenergy.2021.116485

Liu W, Zhao J, Qiu J, et al (2022) Interpretable Hybrid Experimental Learning for Trading Behavior Modeling in Electricity Market. IEEE Trans Power Syst pp 1–1. 10.1109/TPWRS.2022.3173654

Mitarai K, Negoro M, Kitagawa M, et al (2018) Quantum circuit learning. Phys Rev A 98(3):032,309. https://doi.org/10.1103/PhysRevA.98.032309

Niu X, Wang J, Zhang L (2022) Carbon price forecasting system based on error correction and divide-conquer strategies. Applied Soft Computing 118(107):935. https://doi.org/10.1016/j.asoc.2021.107935

Paszke A, Gross S, Massa F, et al (2019) Pytorch: An imperative style, high-performance deep learning library. Adv Neural Inf Process Syst 32

Ruan J, Liang G, Zhao J, et al (2022) An Inertia-based Data Recovery Scheme for False Data Injection Attack. IEEE Trans Ind Inform pp 1–1. https://doi.org/10.1109/TII.2022.3146859

Shor PW (1999) Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. SIAM Rev 41(2):303–332

Sim S, Johnson PD, Aspuru-Guzik A (2019) Expressibility and entangling capability of parameterized quantum circuits for hybrid quantum-classical algorithms. Adv Quantum Tech 2(12):1900,070. 10.1002/qute.201900070, https://arxiv.org/abs/1905.10876 [quant-ph]

Sun W, Huang C (2020) A carbon price prediction model based on secondary decomposition algorithm and optimized back propagation neural network. J Clean Prod 243(118):671. https://doi.org/10.1016/j.jclepro.2019.118671

Sun W, Zhang C (2018) Analysis and forecasting of the carbon price using multi—resolution singular value decomposition and extreme learning machine optimized by adaptive whale optimization algorithm. Applied Energy 231:1354–1371. https://doi.org/10.1016/j.apenergy.2018.09.118

Wendin G (2017) Quantum information processing with superconducting circuits: A review. Rep Prog Phys 80(10):106,001. 10.1088/1361-6633/aa7e1a

Ying X (2019) An overview of overfitting and its solutions. In: J. Phys. Conf. Ser., vol 1168. IOP Publishing, p 022022, https://doi.org/10.1088/1742-6596/1168/2/022022

Zhao H, Zhao J, Qiu J et al (2020) Cooperative Wind Farm Control With Deep Reinforcement Learning and Knowledge-Assisted Learning. IEEE Trans Ind Inform 16(11):6912–6921. https://doi.org/10.1109/TII.2020.2974037

Zhu B, Ye S, Wang P et al (2018) A novel multiscale nonlinear ensemble leaning paradigm for carbon price forecasting. Energy Economics 70:143–157. https://doi.org/10.1016/j.eneco.2017.12.030

Zhu J, Wu P, Chen H et al (2019) Carbon price forecasting with variational mode decomposition and optimalcombined model. Physica A: Statistical Mechanics and its Applications 519:140–158. https://doi.org/10.1016/j.physa.2018.12.017

Zhu B, Chevallier J (2017) Carbon price forecasting with a hybrid Arima and least squares support vector machines methodology. In: Pricing and Forecasting Carbon Markets. Springer, p 87–107

Authors and Affiliations

Yuji Cao[1] · Xiyuan Zhou[2] · Xiang Fei[3] · Huan Zhao[2] · Wenxuan Liu[1,2] · Junhua Zhao[1,2]

Yuji Cao
yujicao@link.cuhk.edu.cn

Xiyuan Zhou
xiyuanzhou1@link.cuhk.edu.cn

Xiang Fei
xiangfei@link.cuhk.edu.cn

Wenxuan Liu
liuwenxuan@cuhk.edu.cn

[1] Center for Crowd Intelligence, Shenzhen Institute of Artificial Intelligence and Robotics for Society (AIRS), Longgang District, Shenzhen 518129, Guangdong, China

[2] School of Science and Engineering, The Chinese University of Hong Kong, Shenzhen, Longgong District, Shenzhen 518172, Guangdong, China

[3] School of Data Science, The Chinese University of Hong Kong, Shenzhen, Longgong District, Shenzhen 518172, China