Expert Systems With Applications 195 (2022) 116583

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

QuantumLeap: Hybrid quantum neural network for financial predictions

Eric Paquet *, Farzan Soleymani

National Research Council, 1200 Montreal Road, Ottawa, ON K1A 0R6, Canada

ARTICLE INFO

Keywords:
Deep quantum neural network
Financial prediction
Regression
Parallel learning
Deep neural network
Hybrid quantum neural network

ABSTRACT

This paper introduces a new hybrid deep quantum neural network for financial predictions, the QuantumLeap system. This system consists of an encoder that transforms a partitioned financial time series into a sequence of density matrices; a deep quantum network that predicts the density matrix at a later time; and a classical network that measures, from the output density matrix, the maximum price reached by a security at a later time. The deep quantum network is isomorphic to a deep classical network and is computationally tractable. A hybrid deep network is associated with each time stride, allowing for parallelisation of the learning process. The classical network is a learnable measurement apparatus which infers, from the output density matrix, the maximum price reached by a security for a given time. Experimental results associated with 24 securities clearly demonstrate the accuracy and efficiency of the system in both the regression and extrapolation regimes.

1. Introduction

As early as the 1980s, quantum computing attracted the interest of researchers for solving complex problems (Nielsen & Chuang, 2002; Schuld et al., 2014). Feynman was the first to suggest that quantum mechanics may outperform classical computations. This eventually led to the application of quantum computing to neural networks (Biamonte et al., 2017). Classical machine learning approaches, and more specifically, deep neural networks (DNNs), have been unquestionably successful in generating and recognising data patterns. In contrast, quantum mechanics has come to be known for its ability to generate and recognise uncommon, although significant, patterns (Biamonte et al., 2017).

More recently, quantum computing has found applications in finance (Egger, Gutierrez et al., 2020). These applications include: the pricing of financial derivatives (Rebentrost et al., 2018), risk evaluation (Woerner & Egger, 2019), portfolio optimisation (Mugel et al., 2020; de Prado, 2015; Rosenberg et al., 2016), asset management (Egger, Gambella et al., 2020), financial crash prediction (Orús et al., 2019), and others. Market crashes have been ubiquitous during the last two decades. The most notable of these is the recession of 2008 subprime mortgage crisis (Farmer, 2012), and the most recent Coronavirus crash (Ashraf, 2020). As a result, the ability to predict financial market behaviour has become of paramount importance (Orús et al., 2019). Empirical and statistical methods have been applied with limited success (Brée et al., 2013).

Financial instruments in the stock market belong to various sectors, including energy, consumer defencive, consumer cyclicals, technology, healthcare, financial services and communication services. Each of these sectors behaves differently according to the financial status of its underlying ecosystem. For instance, since the beginning of the COVID-19 pandemic, the energy sector has been affected by the closure of many production lines, just as transportation companies have been affected by travel restrictions. Likewise, many companies in the technology and consumer sectors such as NVidia, AMD, Tesla, Amazon and Walmart — have experienced unprecedented demand, leading to increases in the value of their stocks. Additionally, over the past decade, with the emergence of cryptocurrencies and companies mining Bitcoin and Ethereum, a new sector has been added to the stock market that has attracted many investors and has shown unprecedented volatility. It can be concluded that the financial instruments are behaving differently. Thus, our approach aims to predict this behaviour using a quantum computing approach (He et al., 2020).

Quantum mechanics has the ability to model the temporal evolution of financial instruments as a quantum process (Haven, 2002). Many financial problems may be formulated in terms of non-convex optimisation (Cornuejols & Tütüncü, 2006). Solving these problems is often computationally prohibitive using classical techniques, but feasible with quantum approaches, such as quantum annealing (Farhi et al., 2000). Alternatively, patterns for making financial predictions may be extracted from historical data using quantum machine learning techniques, Orús et al. (2019) and Rebentrost et al. (2018). However, the computational cost associated with most of these methods severely

* Corresponding author.
E-mail addresses: Eric.Paquet@nrc-cnrc.gc.ca (E. Paquet), Fsole078@uottawa.ca (F. Soleymani).

https://doi.org/10.1016/j.eswa.2022.116583
Received 15 February 2021; Received in revised form 22 November 2021; Accepted 17 January 2022
Available online 31 January 2022

restricts their applications. Reliable financial predictions may substantially improve the return on investment in the stock market. However, this problem is computationally expensive for classical machine learning methods (Zhang et al., 2019).

Quantum computers, on the other hand, are based on qubits, which are expressed as a superposition of states $|0\rangle$ and $|1\rangle$, resulting in an exponential computational boost and enabling simultaneous processing of more inputs in fewer operations compared to classical computers (Elzerman et al., 2005). Quantum algorithms have proven their superiority compared to classical algorithms in many applications, such as Fourier transforms, solving linear equations, and Eigen systems (Adcock et al., 2015). In addition, quantum algorithms may improve the performance of numerous machine learning approaches such as k-means clustering (Lloyd et al., 2013), principal component analysis (PCA) (Lloyd et al., 2014), support vector machines (Rebentrost et al., 2014), as well as deep learning methods, such as Boltzmann machines (Wiebe et al., 2014). Additionally, using wave functions, quantum computing methods can generate distributions with fewer gate operations, which is not feasible with classical computers (Liu et al., 2019). Quantum algorithms fall short, however, in implementation of nonlinearities with quantum unitaries (Schuld et al., 2014). To address this issue, a classical neural network may be implemented along with a quantum neural network (Liu et al., 2019).

This paper presents a new hybrid deep quantum neural network for the prediction of financial trajectories. This system consists of a deep quantum neural network for quantum prediction and a deep classical neural network (Ma & Mao, 2020), which acts as a measurement apparatus that extracts security prices from predicted quantum density matrices. The deep quantum network employs a new quantisation scheme for financial time series and is computationally tractable in terms of the number of hidden layers.

The paper is structured as follows: First, the architecture of the deep quantum neural network, as well as the training method, will be described in Section 2; in Section 3, a new approach for financial trajectories quantisation will be presented, as well as the classical deep neural network that is employed for measurements; Section 4 will follow with the experimental results for 24 NASDAQ stocks; Sections 5 and 6 conclude the paper.

2. Deep quantum neural network

The quantum network consists of multiple quantum perceptrons (Egger, Gambella et al., 2020). Each perceptron corresponds to an arbitrary unitary operator (Elliott, 2002) with m input qubits and n output qubits, resulting in $(2^{m+n})^2-1$ complex parameters. These operators are reminiscent of classical neural network weight matrices (Ma & Mao, 2020). The states are represented by density matrices (Biamonte et al., 2017). In this case, both the input and output density matrices are pure states (Adcock et al., 2015) which means that:

$$\rho^{\text{in}} = \left|\psi^{\text{in}}\right\rangle \left\langle\psi^{\text{in}}\right|, \quad \rho^{\text{out}} = \left|\psi^{\text{out}}\right\rangle \left\langle\psi^{\text{out}}\right| \tag{1}$$

The quantum network consists of an input layer (the input density matrix), L hidden layers (the unitary operators), and an output layer ($L+1 \equiv$ out) (the output density matrix). Following (Beer et al., 2020), our deep quantum network is defined as,

$$\rho^{\text{out}} = \text{tr}_{l\neq\text{out}}\left[\mathcal{U}(\rho^{\text{in}} \otimes \left|0_{\text{out}}, 0_L, \ldots, 0_1\right\rangle\left\langle 0_{\text{in}}, 0_1, \ldots, 0_L, 0_{\text{out}}\right|)\mathcal{U}^\dagger\right] \tag{2}$$

where the trace is taken over all layers apart from the output layer, $\left|0, \ldots, 0\right\rangle$ is the ground state (Chiribella et al., 2008), and $\mathcal{U}$ is the resulting unitary transformation or quantum circuit.

$$\mathcal{U} = \mathbf{U}^{\text{out}}\mathbf{U}^{(L)}\mathbf{U}^{(L-1)} \ldots \mathbf{U}^{(1)} \tag{3}$$

$\left\{\mathbf{U}^{(l)}\right\}_{l=1}^{L}$ are unitary operators ($\mathbf{U}^{(l)}\mathbf{U}^{(l)\dagger} = \mathbf{U}^{(l)\dagger}\mathbf{U}^{(l)} = \mathbb{I}, \quad \forall\ l$). The symbol $\dagger$ represents the Hermitian transpose (transpose of the complex conjugate). Each unitary operator may be factorised as a product of non-commuting unitary operators (Chiribella et al., 2008):

$$\mathbf{U}^{(l)} = \prod_{j=1}^{m_l} \mathbf{U}_j^{(l)} \tag{4}$$

As a result, the network may be expressed as the composition of a sequence of completely positive transition maps (Müller-Hermes & Reeb, 2017),

$$\rho^{\text{out}} = \mathcal{E}^{\text{out}}(\mathcal{E}^{(L)}(\ldots \mathcal{E}^{(2)}(\mathcal{E}^{(1)}(\rho^{\text{in}}))\ldots)) \tag{5}$$

where the transition map over layer l is equal to

$$\mathcal{E}^{(l)}(X^{(l-1)}) \overset{=}{\wedge} \text{tr}_{l-1}\left[\prod_{j=m_l}^{1} \mathbf{U}_j^{(l)}\ \left(\ X^{(l-1)} \otimes \left|0_l, \ldots, 0_1, 0_{\text{in}}\right\rangle\right. \right. \tag{6}$$
$$\left.\left.\left\langle 0_{\text{in}}, 0_1, \ldots, 0_L, 0_{\text{out}}\right|\ \right) \prod_{j=1}^{m_l} \mathbf{U}_j^{(l)\dagger}\right]$$

and the trace being taken over layer $l-1$. The structure of Eq. (5) is reminiscent of a dense deep neural network in which each transition map may be assimilated to an activation function. The training cost function is based on the fidelity (Beer et al., 2020) which is essentially the sole measure for pure states, that evaluates the closeness between the network output ρ_t^{out} and the true output $\left|\psi_t^{\text{out}}\right\rangle$,

$$\mathcal{L} = \frac{1}{t_f - t_i + 1}\sum_{t=t_i}^{t_f} \left\langle\psi_t^{\text{out}}\right| \rho_t^{\text{out}} \left|\psi_t^{\text{out}}\right\rangle, \quad \mathcal{L} \in [0,1] \tag{7}$$

where t is a particular instance of the training set (the time series at time t). The cost function varies between 0 and 1, with the latter occurring when the quantum network output is equal to that of the true output. During the training phase, the unitary operators are updated according to

$$\mathbf{U}_j^{(l)} \rightarrow \exp[i\varepsilon\mathbf{K}_j^{(l)}]\mathbf{U}_j^{(l)}, \quad i \equiv \sqrt{-1} \tag{8}$$

which ensures that they remain unitary (Elliott, 2002). After a training iteration, the cost function (Beer et al., 2020) varies as

$$\Delta C = \frac{\varepsilon}{t_f - t_i + 1}\sum_{t=t_i}^{t_f}\sum_{l=1}^{L+1} \text{tr}[\sigma_t^{(l)}\ \Delta\mathcal{E}^{(l)}(\rho_t^{(l-1)})] \tag{9}$$

where

$$\rho_t^{(l)} = \mathcal{E}^{(l)}(\ldots \mathcal{E}^{(2)}(\mathcal{E}^{(1)}(\rho_t^{\text{in}}))\ldots) \tag{10}$$

is the density matrix for layer l,

$$\sigma_t^{(l)} \overset{\wedge}{=} \mathcal{F}^{(l+1)}(\ldots \mathcal{F}^{(L)}(\mathcal{F}^{\text{out}}(\left|\psi_t^{\text{out}}\right\rangle\left\langle\psi_t^{\text{out}}\right|))\ldots) \tag{11}$$

and $(\mathcal{F})$ is the adjoint channel of the completely positive map $(\mathcal{E})$:

$$\mathcal{F}^{(l)}(X^{(l-1)}) = \text{tr}_{l-1}\left[\prod_{j=m_l}^{1} \mathbf{U}_j^{(l)\dagger}\ \left(\ \left|0_{\text{out}}, 0_L, \ldots, 0_1, 0_{\text{in}}\right\rangle\right. \right. \tag{12}$$
$$\left.\left.\left\langle 0_{\text{in}}, 0_1, \ldots, 0_l\right| \otimes X^{(l-1)}\ \right) \prod_{j=1}^{m_l} \mathbf{U}_j^{(l)}\right]$$

The training algorithm is outlined in Eq. (13a) (Beer et al., 2020). First, an initial unitary operator is chosen randomly. Second, for all layers and training pairs, the density matrix is evaluated recursively, layer by layer. Third, the parameter matrix $\mathbf{K}_j^{(l)}$ is calculated, with the trace being taken over all qubits unaffected by $\mathbf{U}_j^{(l)}$ and η, being the learning rate (Beer et al., 2020). Finally, the unitary operator is updated. Steps 2 and 3 are repeated until the cost function reaches a maximum (ideally one). As a result, the parameter matrix may be evaluated layer by layer. Only two layers are required at once, thus considerably reducing the amount of memory required. Consequently, the dimensions of the matrices scale with the number of layers, thus ensuring the deep quantum neural networks are computationally tractable.

$$\mathbf{U}_j^{(l)}, \quad j \sim p(j), \quad l \sim p(l), \tag{13a}$$

$$\forall[(\left|\psi_t^{\text{in}}\right\rangle,\left|\psi_t^{\text{out}}\right\rangle)\in\left\{(\left|\psi_t^{\text{in}}\right\rangle,\left|\psi_t^{\text{out}}\right\rangle)\right\}_{t=t_i}^{t_f}]\wedge\forall\,[l]\Rightarrow\rho_t^{(l)}=\text{tr}_{l-1}\mathcal{E}^l\left(\rho_t^{(l-1)}\right)$$

$$=\text{tr}_l-1\left[\prod_{j=m_l}^{1}\mathbf{U}_j^{(l)}\left(\rho_t^{(l-1)}\otimes\left|0_{\text{out}},0_L,\ldots,0_1\right\rangle\right.\right. \tag{13b}$$

$$\left.\left.\left\langle 0_{\text{in}},0_1,\ldots,0_L,0_{\text{out}}\right|\right)\prod_{j=1}^{m_l}\mathbf{U}_j^{(l)\dagger}\right],$$

$$\mathbf{U}_j^{(l)}\rightarrow\exp\left[i\epsilon\mathbf{K}_j^{(l)}\right]\mathbf{U}_j^{(l)}, \tag{13c}$$

$$\mathbf{K}_j^{(l)}=\eta\frac{2^m_{l-1}}{t_f-t_i+1}\sum_{t=t_i}^{t}\text{tr}_{\not\subset\mathbf{U}_j^{(l)}}\circ\mathbf{M}_j^{(l)}, \tag{13d}$$

where

$$\mathbf{M}_j^{(l)}=\left[\prod_{\alpha=j}^{1}\mathbf{U}_\alpha^{(l)}(\rho_t^{(l-1;\,l)})\prod_{\alpha=1}^{j}\mathbf{U}_\alpha^{(l)\dagger}\times\prod_{\alpha=j+1}^{m_l}\mathbf{U}_\alpha^{(l)\dagger}(\mathbb{I}_{l-1}\otimes\sigma_t^{(l)})\prod_{\alpha=m_l}^{j+1}\mathbf{U}_\alpha^{(l)}\right] \tag{14a}$$

$$\rho_t^{(l-1;\,l)}\triangleq\rho_t^{(l-1)}\otimes\left|0_{\text{out}},0_L,\ldots,0_1\right\rangle\left\langle 0_{\text{in}},0_1,\ldots,0_L,0_{\text{out}}\right|, \tag{14b}$$

$$\sigma_t^{(l)}=\mathcal{F}^{(l+1)}(\ldots\mathcal{F}^{\text{out}}(\left|\psi_t^{\text{out}}\right\rangle\left\langle\psi_t^{\text{out}}\right|)\ldots) \tag{14c}$$

The network architecture is outlined in Fig. 1

3. Financial trajectories quantisation and measurement

A financial trajectory or time series may be associated with any security. This time series may be partitioned into a sequence of discrete segments with a sliding window. Assuming a window size of $\Delta=2,4,\ldots$ and a time interval of one trading day, the partitioned time series becomes:

$$\left[S_{t_i},S_{t_i+1},\ldots,S_{t_f}\right]\underset{w_\Delta}{\rightarrow} \tag{15}$$

$$\left[(S_{t_i},\ldots S_{t_i+\Delta-1}),(S_{t_i+1},\ldots S_{t_i+\Delta}),\ldots,(S_{t_f-(\Delta-1)},\ldots S_{t_f})\right]$$

Here, t refers to a trading day. The window size must be a multiple of two in order to produce qubit state vectors. The hybrid deep quantum network, named QuantumLeap, consists of an encoder that creates for each window, a state vector and a density matrix; a deep quantum network that predicts the density matrix at a later time; and a classical deep neural network, that measures the maximum price reached by a security during a trading day, from the output density matrix. The encoder associates a complex state vector to every window,

$$(S_t,S_{t+1},\ldots,S_{t+\Delta-1})\mapsto\left|\psi_{S_t}^{\text{in}}\right\rangle \tag{16}$$
$$=|\min S_{t+(\Delta-1)}+i\max S_{t+(\Delta-1)},$$
$$\ldots,\min S_{t+1}+i\max S_{t+1},\min S_t+i\max S_t\rangle$$

where is the minimum price (low) reached by the security during a trading day, while is its maximum (high). Minimum prices are never measured by the classical networks; they act as hidden parameters or constraints for the quantum network (Elliott, 2002). The corresponding input density matrix is evaluated with Eq. (1). The architecture of this deep quantum neural network is defined by Eq. (1) to (6) and resembles the architecture of a dense deep neural network. The quantum network is trained according to Eq. (13a) in order to maximise the cost in Eq. (7). The state vector associated with the output layer is

$$\left|\psi_{S_{t+\tau}}^{\text{out}}\right\rangle=|\min S_{t+(\Delta-1)+\tau}+i\max S_{t+(\Delta-1)+\tau},\ldots \tag{17}$$
$$,\min S_{t+1+\tau}+i\max S_{t+1+\tau},\min S_t+\tau+i\max S_t+\tau\rangle\ ,\ \tau>\Delta w$$

Where τ is the stride (the time at which the security price is predicted). The corresponding density matrix is evaluated with Eq. (1). Unfortunately, these state vectors are not normalised and a standard normalisation would result in too great a loss of information. Nonetheless, it has been demonstrated that any complex field has an equivalent normalised expression of the form:

$$\left|\psi_{S_t}\right\rangle\mapsto\left|\psi_{S_t}\right\rangle=\left|(\exp[i\theta_{t+\Delta-1}]+\exp[i\theta'_{t+\Delta-1}]),\right. \tag{18}$$
$$\left.\ldots,(\exp[i\theta_{t+1}]+\exp[i\theta'_{t+1}]),(\exp[i\theta_t]+\exp[i\theta'_t])\right\rangle$$

where

$$\theta_t=\varphi_t+\arccos(\frac{A_t}{A_{\max}}), \tag{19a}$$

$$\theta'_t=\varphi_t-\arccos(\frac{A_t}{A_{\max}}), \tag{19b}$$

$$A_t=\left\|\min S_t+i\max S_t\right\|, \tag{19c}$$

$$\varphi_t=\arg[\min S_t+i\max S_t] \tag{19d}$$

The output of the quantum network is a complex density matrix. A measurement must be performed on this matrix in order to extract a real quantity of interest. The measurement apparatus is a classical dense deep neural network which measures the price of the security at time $t+\tau$. The classical network is trained independently from the quantum network with the adaptive moment estimation algorithm (Adam). In this work, the aim is to predict the price of a security over a period of twenty days. However, rather than employing a single hybrid quantum network for the predictions, we train $\tau_{\max}$ identical hybrid quantum networks, one for each stride, reducing the complexity of the learning process. The approach may be schematised as follows:

$$[S_{t_i},S_{t_i+1},\ldots,S_{t_f}]\underset{w_\Delta}{\Rightarrow} \tag{20a}$$
$$(S_{t_i},\ldots S_{t_i+\Delta-1}),(S_{t_i+1},\ldots S_{t_i+\Delta}),\ldots,(S_{t_f-(\Delta-1)},\ldots S_{t_f})$$

$$\left\{\left|\psi_{S_t}^{\text{in}}\right\rangle\right\}_{t=t_i}^{t_f}=\{\ \psi_{S_t}^{\text{in}}=\left|\min S_{t+(\Delta-1)}+i\max S_{t+(\Delta-1)},\right. \tag{20b}$$
$$\left.\ldots,\min S_{t+1}+i\max S_{t+1},\min S_t+i\max S_t\right\rangle\ \}_{t=t_i}^{t_f}\Rightarrow$$

$$\left\{\left|\psi_{S_{t+\tau}}^{\text{out}}\right\rangle\right\}_{t=t_f}^{t_f}=\{\ \psi_{S_{t+\tau}}^{\text{out}}=\left|\min S_{t+(\Delta-1)+\tau}+i\max S_{t+(\Delta-1)+\tau},\right. \tag{20c}$$
$$\left.\ldots,\min S_{t+1+\tau}+i\max S_{t+1+\tau},\min S_{t+\tau}+i\max S_{t+\tau}\right\rangle\ \}_{t=t_f}^{t_f}\Rightarrow$$

$$\left\{\ \left|\psi_{S_t}\right\rangle\mapsto\left|\psi_{S_t}\right\rangle=\left|\exp[i\theta_{t+\Delta-1}]+\exp[i\theta'_{t+\Delta-1}],\right.\right. \tag{20d}$$
$$\left.\left.\ldots,\exp[i\theta_{t+1}]+\exp[i\theta'_{t+1}],\exp[i\theta_t]+\exp[i\theta'_t]\right\rangle\ \right\}_{t=t_i}^{t_f}\Rightarrow$$

$$\left\{\rho_{S_t}^{\text{in}}=\left|\psi_{S_t}^{\text{in}}\right\rangle\left\langle\psi_{S_t}^{\text{in}}\right|\right\}_{t=t_i}^{t_t}, \tag{20e}$$

$$\left\{\rho_{S_{t+\tau}}^{\text{out}}=\left|\psi_{S_{t+\tau}}^{\text{out}}\right\rangle\left\langle{}_{S_{t+\tau}}^{\text{out}}\right|\right\}_{t=t_i}^{t_f},$$

$$\Rightarrow\ X_\tau=\left\{(\rho_{S_t}^{\text{in}},\rho_{S_{t+\tau}}^{\text{out}})\right\}_{t=t_i}^{t_f},$$

where

$$X_\tau\Rightarrow\left\{\rho_{S_{t+\tau}}^{\text{out}}=\mathcal{E}^{\text{out}}(\mathcal{E}^L(\ldots\mathcal{E}^2(\mathcal{E}^1(\rho_{S_t}^{\text{in}}))\ldots))\right\}_{t=t_i}^{t_f}\ ,\ \tau=1,\ldots,\tau_{\max} \tag{21a}$$

$$\left\{(\rho_{S_{t+\tau}}^{\text{out}},\max S_{t+\tau})\right\}_{t=t_i}^{t_f}\Rightarrow\left\{\max S_{t+\tau}=\xi_\tau^{\ell}(\ldots\xi_\tau^1(\rho_{S_{t+\tau}}^{\text{out}};\omega_1)\ldots;\omega_\ell)\right\}_{t=t_i}^{t_f},$$
$$\tau=1,\ldots,\tau_{\max}$$

First, the financial time series is partitioned with a sliding window of size Δ. Second, the unnormalised input and output state vectors are created. They are normalised according to Eq. (18), and density matrices are evaluated, and the training set is created. Third, the deep quantum neural networks are trained using the training set according to Eq. (13a). Each network consists of one input layer, L hidden layers, and one output layer. The input and the output layers are density matrices, while the hidden layers are unitary operators. There are a total of $\tau_{\max}$ quantum networks corresponding to $\tau_{\max}$ strides. Fourth, the corresponding $\tau_{\max}$ deep classical neural networks are trained with the semi-classical training set $\left\{(\rho_{S_{t+\tau}}^{\text{out}},\max S_{t+\tau})\right\}$ (the classical element being the maximum price). Therefore, for each quantum network, there is a classical network that acts as a measurement apparatus. Each classical network has *ell* hidden layers, each of which is parameterised by ω_k (weights and biases); the activation functions are

$$\xi_\tau^i,\quad\tau=1,\ldots,\tau_{\max},\quad i=1,\ldots,L \tag{22}$$

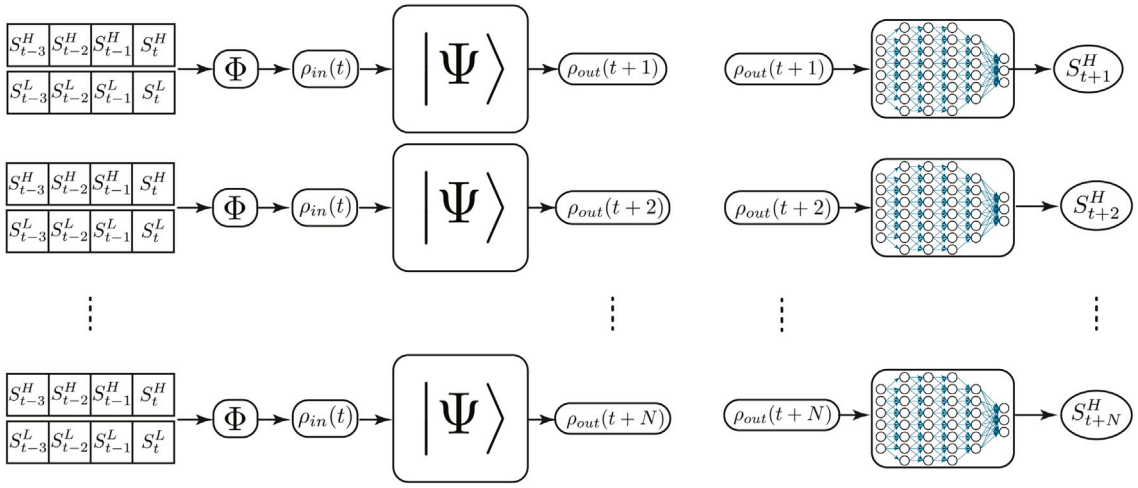

Fig. 1. Outline of QuantumLeap's architecture. The left schema shows the time window for the low and high values of a stock, the encoder (Eq. (19)), the input density matrix, the quantum neural network and the output density matrix. The right schema shows the output density matrix, the deep dense neural network and the prediction for the stock's high value. It should be noted that each time strike τ has its own dedicated neural networks, both quantum and classical.

Table 1
Hyperparameters of the deep quantum neural networks.

Parameter	Value
L (number of hidden layers)	5
Dimensions of unitary operators	Eq. (23)
ϵ (cf. Eq. (8))	0.1
η (learning rate)	1
Number of epochs	20
τ_{max} (strides and number of networks)	20
Δ (window size)	4

Table 2
Hyperparameters for the deep classical neural networks.

Parameter	Value
ℓ (number of hidden layers)	4
β_1 (Adam)	0.9
β_2 (Adam)	0.999
ϵ (Adam)	0.00001
τ_{max} (strides and number of networks)	20
Δ (window size)	4

4. Experimental results

The 24 stocks that made up the data set were Apple (AAPL), Amazon (AMZN), Boeing (BA), Bank of America (BAC), British Petroleum (BP), Caterpillar (CAT), Cisco (CSCO), General Electric (GE), Alphabet (GOOGL), Home Depot (HD), International Business Machines (IBM), Juniper Networks (JNJ), JPMorgan (JPM), Nordstrom (JWN), Coca-Cola (KO), Manulife Financial (MFC), 3M (MMM), Merck (MRK), Microsoft (MSFT), Oracle (ORCL), Pfizer (PFE), Royal Dutch Shell (RDSB), Toronto Dominion (TD), and Walmart (WMT). Most of these stocks are traded on the NASDAQ. These stocks encompass various sectors, such as high-tech, energy, retail, e-commerce, and pharmaceuticals, among others. The data spans a time period ranging from 1 January 2019 to 11 September 2020. The experiment aims at predicting share prices both in the regression and extrapolation regimes, the former being associated with the training data set. The training set consists of the first 407 trading days while the remaining 20 days form the test set (the days that must be predicted). As a result, the last output window of the training set is $[S_{404}, S_{405}, S_{406}, S_{407}]$. The quantum neural networks were trained according to Eqs. (13a) and (20) for 20 epochs while their classical counterparts were trained with Adam. The hyperparameters for the quantum and classical networks are reported in Tables 1 and 2, respectively.

The dimensions of the unitary operators were:

$$\begin{cases} \dim U_1^{(1)} = \dim U_2^{(1)} = 32 \times 32 \\ \dim U_1^{(2)} = \dim U_2^{(2)} = \dim U_3^{(2)} = 32 \times 32 \\ \dim U_1^{(3)} = \dim U_2^{(3)} = \dim U_3^{(3)} = \dim U_4^{(3)} = 128 \times 128 \\ \dim U_1^{(4)} = \dim U_2^{(4)} = \dim U_3^{(4)} = 128 \times 128 \\ \dim U_1^{(5)} = \dim U_2^{(5)} = 32 \times 32 \end{cases} \tag{23}$$

The training was performed on an HP Apollo System with 20 Dual Intel Xeon Gold 6149 processors and 968 GB of memory. The total duration of the calculations for all 24 stocks was approximatively 24 h. A total of $24 \times 20 = 480$ were trained corresponding to the 24 stocks and the 20 strides. They were all trained in parallel in batches of 20. The results for both the regression and the extrapolation are reported in Figs. 2 to 6 with stride equal to 1, 5, 10, 15, and 20 days, respectively, with the extrapolation spanning the last 20 days.

These experimental results clearly demonstrate the ability of QuantumLeap to predict stock prices over a relatively long period of time. The prediction relative error for extrapolation is reported in Fig. 7. Each curve consists of 20 trading days. Each daily prediction is made by a different hybrid quantum network (one for each day [or stride]).

These results are further reported in Tables 3 and 4

With the exception of Apple, Boeing and Cisco, for which the maximum error is 8%, the relative error is usually far below 1%. These figures clearly demonstrate the ability of QuantumLeap to make accurate predictions for a plurality of securities from various industry sectors. The fact that the relative errors over the prediction is very low, also indicates that there is no over-fitting, despite the fact that the regressions have a very high accuracy.

To further validate the performance of our approach, we compared our quantum neural network with two classical models: a classical implementation of our quantum network and the autoregressive integrated moving average (ARIMA) method, which is commonly employed for time series and financial predictions (Öztürk & Wolfe, 2000; Zieliński, 1990). The best parameters for ARIMA were estimated using the maximum likelihood method (Almasarweh & Alwadi, 2018; Ariyo et al., 2014; Meyler et al., 1998). The three models were compared using the root mean square value (RMSV) metric. The results (reported in Table 5) clearly show that our quantum approach outperforms its classical counterparts. To highlight the distinction between the quantum and classical models, we compared RMSV distributions using a non-parametric *Mann–Whitney–Wilcoxon* test (Öztürk & Wolfe, 2000; Zieliński, 1990). The results are reported in Table 6. While the classical

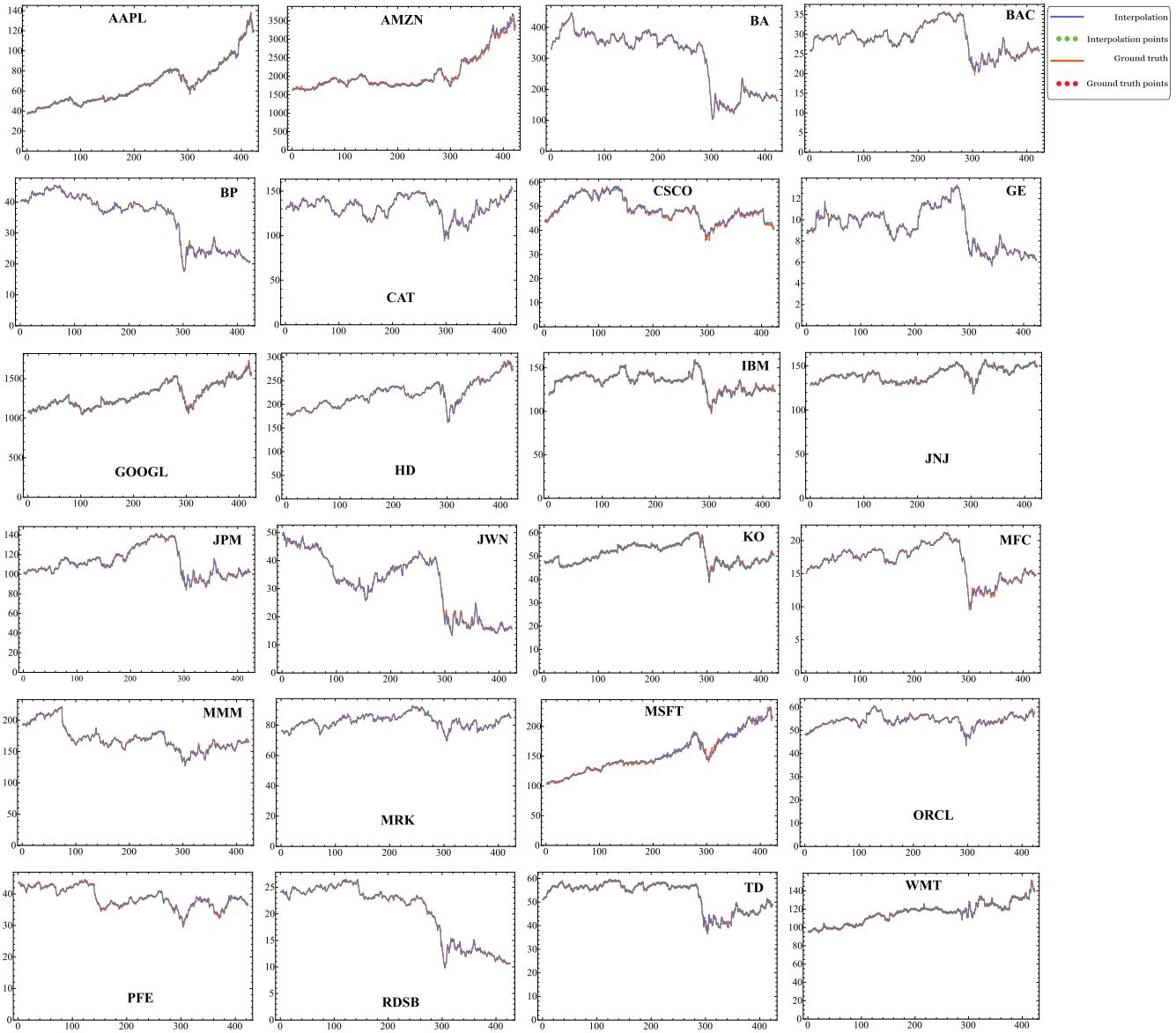

Fig. 2. Regressions and predictions (last 20 points) for $\tau = 1$. The market and predicted prices are represented by the blue and orange lines, respectively.

distributions are comparable ($p = 0.94$), they clearly differ from the quantum model, with p-values of 0.000066 for the classical network and 0.000093 for ARIMA.

5. Discussion

The explanation for the quality of the results may be found in the very nature of the network. The quantum network is entirely based on quantum mechanics, which means that it is more likely to learn quantum behaviours than a classical network. For instance, in quantum mechanics, the temporal evolution is associated with a unitary operator. While present in our quantum network, these operators are totally absent from classical networks. Therefore, we postulate that financial data dynamics could be potentially formulated in terms of quantum mechanics. Indeed, the behaviour of a security may be modelled by a stochastic equation (Baaquie, 2009):

$$\frac{dS(t)}{dt} = \mu(t)S(t) + \sigma(t)S(t)\xi(t) : \begin{cases} E[\xi(t)] = 0, \\ E[\xi(t)\xi(t')] = \delta(t-t'), \\ 1 \le t, t' \le t_f \end{cases} \tag{24}$$

where $\mu(t)$ is the drift, $\sigma(t)$ is the volatility, $\xi(t)$ is the random noise and, $E[\cdot]$ is the mathematical expectation. From Eq. (24) and without making any assumption, one may demonstrate (Baaquie, 2009) that the expectation of the security is given by

$$\begin{aligned} E[z] \triangleq \int \mathcal{D}z\ z\ \Pr[z] &= \frac{1}{\int \mathcal{D}z\ \exp\left[\mathcal{A}[z]\right]} \\ &\times \left[\prod_{t=t_i}^{t_f} \int_{-\infty}^{\infty} dz(t)\ \ z(t) \exp\left[-\mathcal{A}[z]\right]\right] : \\ \mathcal{A}[z] &= \frac{1}{2}\int_{t_i}^{t_f} \left[\frac{\frac{\partial z(t)}{\partial t} + \mu(t)}{\sigma(t)}\right]^2 dt, z(t) \equiv \ln S(t) \end{aligned} \tag{25}$$

The above equation represents a Feynman–Dirac path integral (Masujima, 2008). The Feynman–Dirac path integral is an equivalent formulation of the Schrödinger equation and therefore of quantum mechanics. It evaluates a weighted sum for the stock price, overall possible financial trajectories. A probability is associated with each price. This probability is expressed in terms of a Euclidean action (Masujima, 2008) $\mathcal{A}[z]$, with the latter being the time integral of the Lagrangian (Masujima, 2008) associated with Eq. (25). Therefore, the

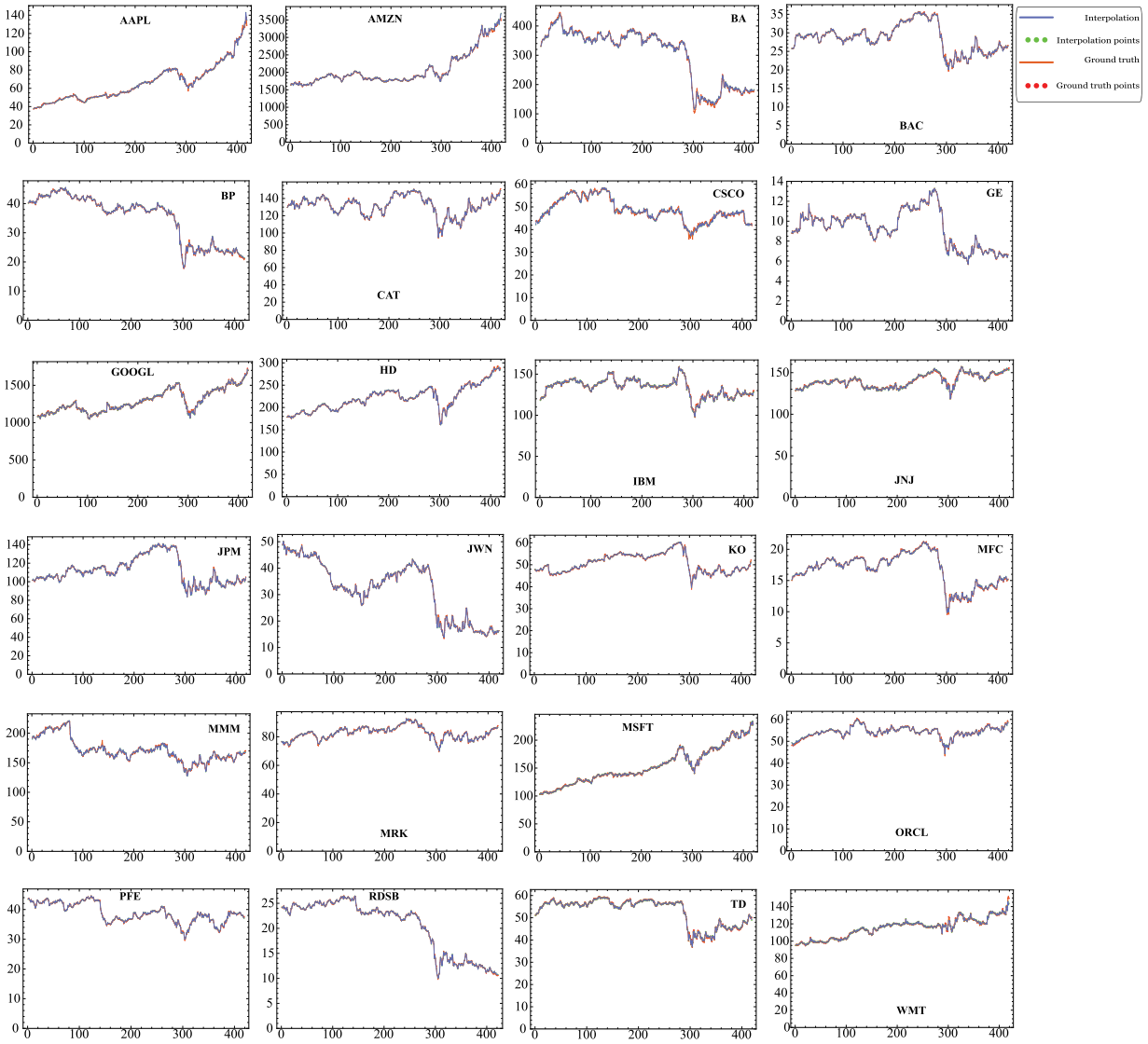

Fig. 3. Regressions and predictions (last 20 points) for $\tau = 5$. The market and predicted prices are represented by the blue and orange lines, respectively.

temporal evolution of a security is essentially quantum in nature and as such, may be better modelled with a quantum neural network. This approach will be expanded upon in the future to include risk management and, more specifically, pandemic evolution prediction. The method will also be extended to other financial instruments, such as options, forward and future contracts, and swaps, as well as to consider multiple financial instruments simultaneously, with mixed state density matrices (Cohen-Tannoudji et al., 2005).

6. Conclusion

A new hybrid deep quantum neural network for financial predictions was introduced. The QuantumLeap system consists of an encoder that transforms a partitioned financial time series into a sequence of density matrices; a deep quantum network that predicts the density matrix at a later time; and a classical network that measures the maximum price reached by the security at a later time, from the output density matrix. The deep quantum network is isomorphic to a deep classical neural network and is computationally tractable in terms of the number of hidden layers. A hybrid deep network is associated with each time stride, allowing for parallelisation of the learning process. The classical network is a learnable measurement apparatus which infers, from the output density matrix, the maximum price reached by a security at a given time. The experimental results, based on 24 securities, clearly demonstrate the accuracy and the efficiency of the system both in the regression and the extrapolation (prediction) regimes. The former spanned 407 days while the latter was restricted to 20 days. QuantumLeap applications include hedging, risk mitigation, and portfolio management.

CRediT authorship contribution statement

Eric Paquet: Conception and design of study, Acquisition of data, Analysis and/or interpretation of data, Writing – original draft, Writing – review & editing. **Farzan Soleymani:** Conception and design of study, Acquisition of data, Analysis and/or interpretation of data, Writing – original draft, Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

All authors approved the final version of the manuscript.

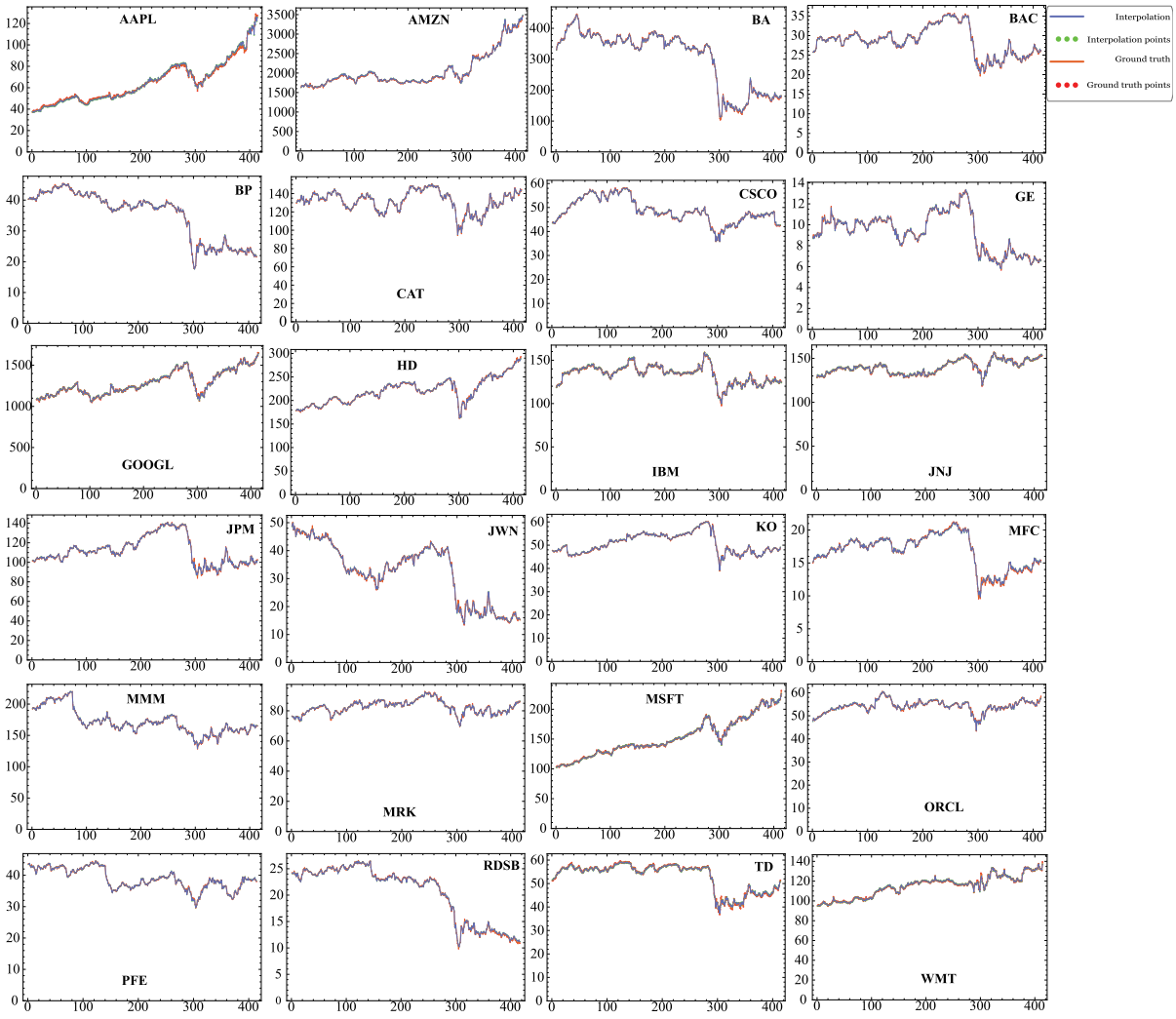

Fig. 4. Regressions and predictions (last 20 points) for $\tau = 10$. The market and predicted prices are represented by the blue and orange lines, respectively.

Table 3
Relative errors for QuantumLeap's predictions over a 20-trading-day period. Each daily prediction is performed with a different hybrid quantum network (Day 1 to 9).

Stocks	Day - 1	Day - 2	Day - 3	Day - 4	Day - 5	Day - 6	Day - 7	Day - 8	Day - 9
AAPL	0.25	1.06	2.33	0.63	2.45	0.70	1.39	2.87	1.47
AMZN	2.68	1.43	0.75	0.32	0.98	0.96	1.15	2.01	1.46
BA	1.32	3.39	1.30	0.79	1.05	1.22	2.98	0.12	1.63
BAC	2.22	2.48	1.28	0.64	0.49	0.41	0.41	1.79	1.96
BP	0.24	1.26	2.30	1.57	0.88	1.13	1.09	0.81	1.14
CAT	0.66	1.52	0.54	0.03	0.75	0.25	0.67	0.05	0.26
CSCO	1.64	2.11	0.65	1.89	7.14	2.78	1.20	1.23	3.18
GE	1.78	0.28	0.86	0.57	0.57	2.23	0.71	0.53	0.16
GOOGL	0.19	0.17	0.04	0.81	0.10	0.01	0.17	0.26	0.10
HD	0.38	0.01	0.42	0.03	0.16	0.53	0.35	1.70	0.87
IBM	0.20	1.00	0.18	0.30	0.19	0.88	0.29	0.29	0.35
JNJ	0.06	0.15	0.20	0.60	0.79	0.43	0.00	0.07	0.73
JPM	0.47	0.09	0.01	0.29	0.04	0.04	0.45	0.33	0.00
JWN	0.58	0.43	1.11	0.53	0.75	1.13	4.32	1.35	1.91
KO	0.13	0.03	0.46	0.03	0.22	0.23	0.25	0.24	0.65
MFC	2.06	0.31	0.61	0.10	0.68	0.38	0.89	0.39	0.64
MMM	0.58	0.43	0.24	0.10	0.32	0.80	0.35	0.23	0.09
MRK	0.68	0.04	0.08	0.37	0.87	0.08	0.08	0.04	0.00
MSFT	0.24	0.54	0.01	0.14	1.23	0.74	0.31	0.02	0.23
ORCL	0.16	0.58	0.17	0.16	0.06	0.58	0.27	0.35	0.74
PFE	0.33	0.73	0.12	0.54	0.07	0.37	0.05	0.17	0.10
RDSB	0.20	1.37	2.28	2.64	0.16	0.48	2.72	0.69	0.75
TD	1.20	0.52	0.35	0.38	0.39	0.32	0.71	0.38	1.29
WMT	1.24	0.01	0.44	0.07	0.10	0.28	0.87	0.24	0.84

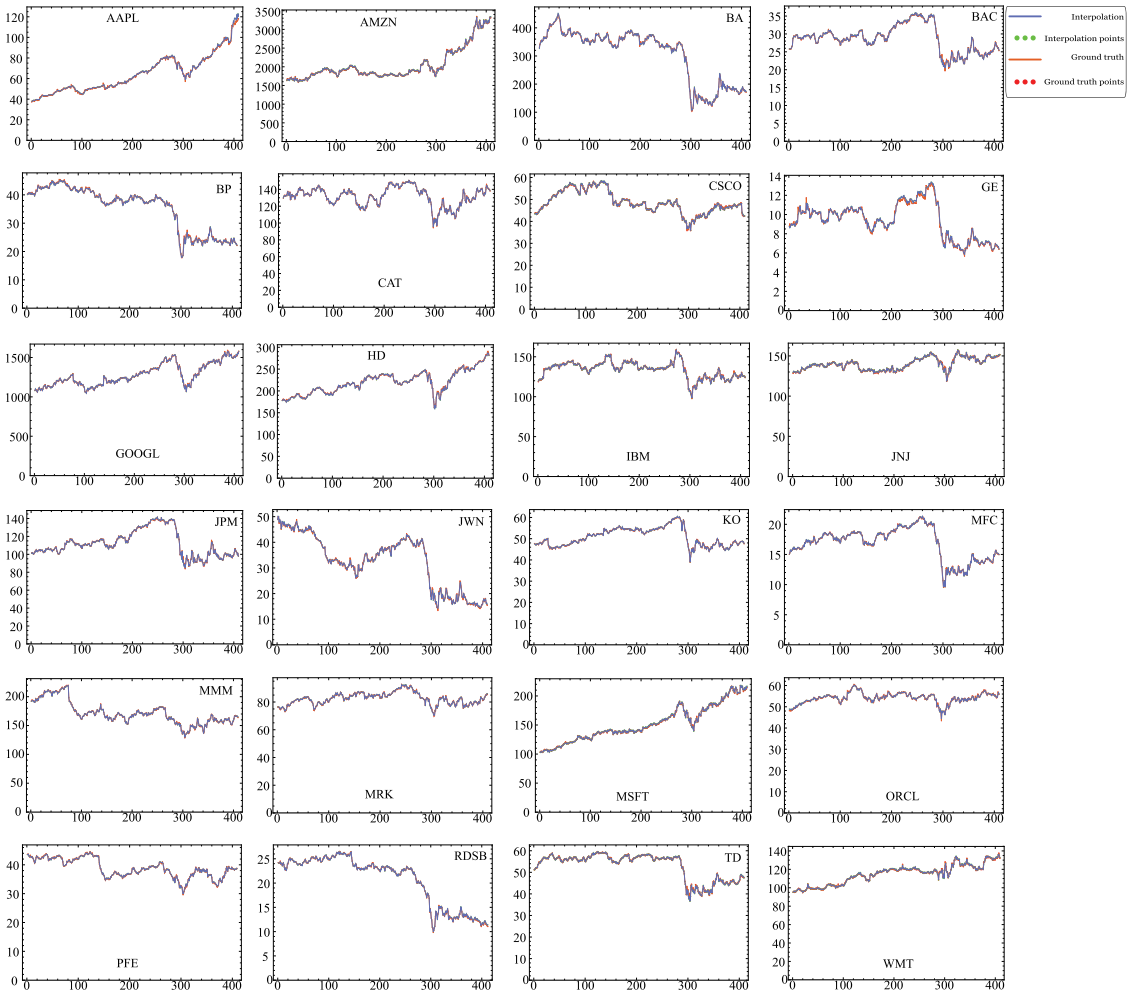

Fig. 5. Regressions and predictions (last 20 points) for $\tau = 15$. The market and predicted prices are represented by the blue and orange lines, respectively.

Table 4
Relative errors for QuantumLeap's predictions over a 20-trading-day period. Each daily prediction is performed with a different hybrid quantum network (Day 10 to 20).

Stocks	Day - 10	Day - 11	Day - 12	Day - 13	Day - 14	Day - 15	Day - 16	Day - 17	Day - 18	Day - 19	Day - 20
AAPL	1.69	4.10	0.16	3.69	0.60	1.16	2.80	0.53	2.31	6.67	0.21
AMZN	1.10	0.11	0.40	0.11	0.36	0.78	0.41	1.66	1.66	0.30	0.68
BA	0.49	1.26	1.27	1.95	0.58	0.52	1.80	0.43	1.30	0.48	0.37
BAC	0.81	0.47	0.70	1.41	0.60	0.72	0.55	0.07	2.13	0.48	0.42
BP	0.04	0.94	1.79	0.15	0.11	0.69	1.05	0.53	0.84	0.40	0.03
CAT	0.80	0.47	0.14	0.40	0.14	0.39	0.18	0.24	0.40	0.45	0.20
CSCO	0.43	5.52	2.39	3.43	0.53	0.75	3.27	2.43	1.74	5.33	3.01
GE	0.82	1.66	0.60	0.93	0.43	1.38	0.72	1.66	0.49	0.68	0.67
GOOGL	0.50	0.09	1.92	0.03	0.20	0.32	0.94	0.03	0.43	0.19	0.55
HD	0.16	1.21	0.17	0.52	1.39	0.59	0.65	0.06	0.60	0.23	0.24
IBM	0.76	0.83	0.37	0.44	0.51	0.55	0.44	0.52	0.86	0.06	0.71
JNJ	0.12	0.10	0.07	0.30	0.11	0.20	0.37	0.15	0.28	0.15	0.28
JPM	2.01	0.36	0.49	0.60	2.00	0.53	0.08	0.43	1.49	0.40	0.24
JWN	1.58	1.31	0.35	3.53	0.62	0.57	0.85	0.87	1.75	0.96	1.33
KO	0.33	0.42	0.14	0.09	0.22	0.14	0.12	0.55	0.15	0.17	0.17
MFC	1.03	1.57	0.80	0.43	0.06	0.84	0.76	0.20	0.77	2.18	0.94
MMM	0.43	0.08	0.13	0.89	0.47	0.44	0.47	0.21	0.17	0.52	0.01
MRK	0.21	0.58	0.07	0.12	0.33	0.25	0.35	0.22	0.22	0.29	0.06
MSFT	0.44	0.25	1.30	0.33	1.12	1.32	0.29	0.81	0.06	0.51	0.13
ORCL	0.49	0.16	0.06	0.30	0.42	0.61	0.28	0.72	0.00	0.27	0.10
PFE	1.14	0.89	0.02	0.82	1.18	0.10	0.23	0.46	0.28	0.38	0.15
RDSB	1.64	2.24	1.02	1.48	0.67	1.90	3.17	0.99	0.66	1.68	1.43
TD	1.96	0.62	0.10	0.32	1.09	1.21	0.04	0.02	0.30	0.82	0.20
WMT	0.24	0.14	0.37	0.25	0.70	0.55	0.99	0.58	0.23	0.37	0.00

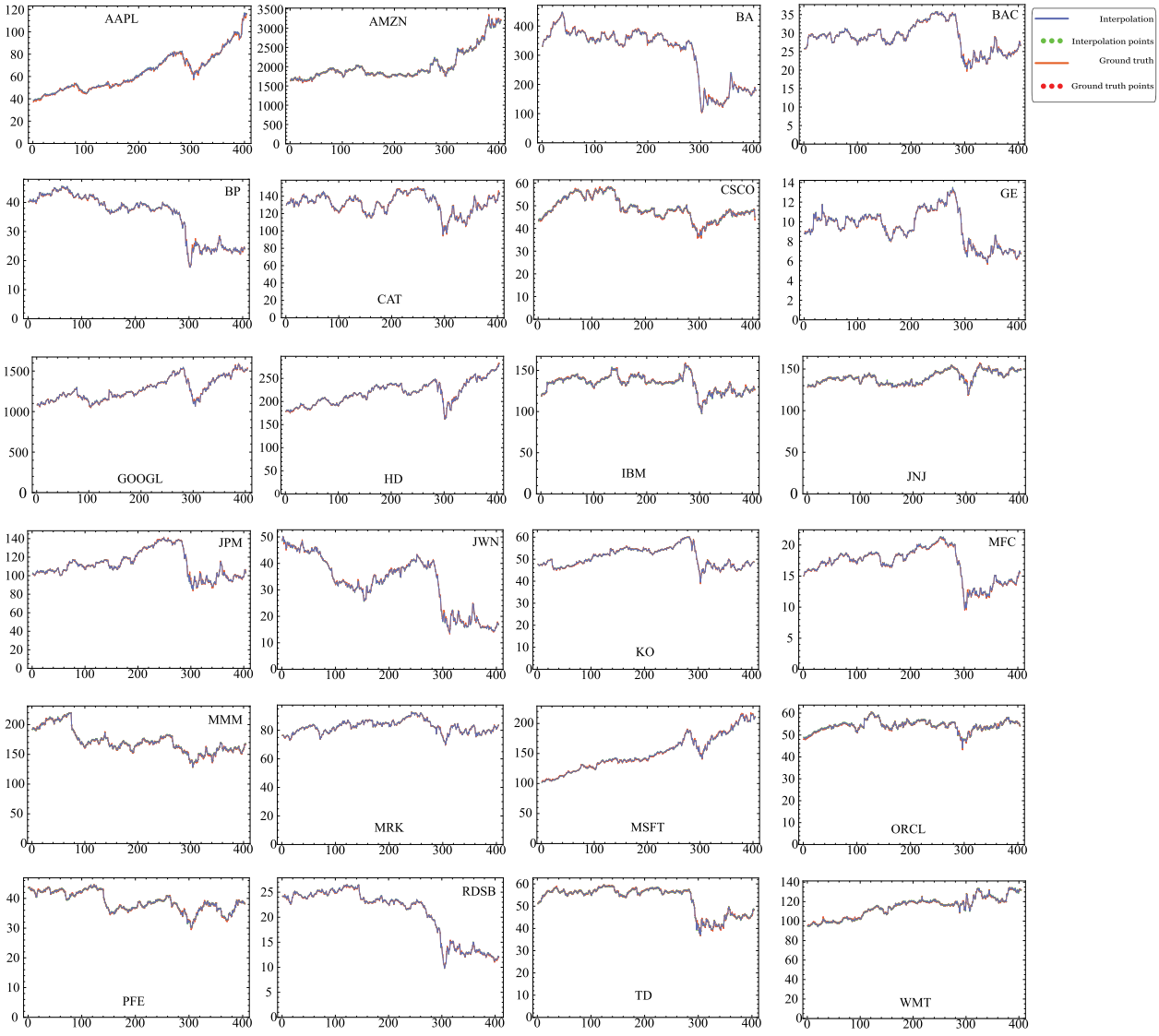

Fig. 6. Regressions and predictions (last 20 points) for $\tau = 20$. The market and predicted prices are represented by the blue and orange lines, respectively.

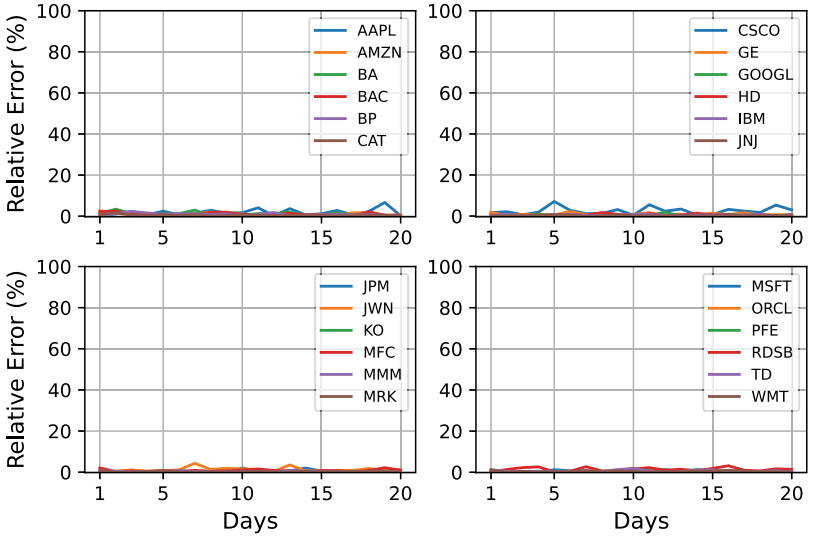

Fig. 7. Relative errors for QuantumLeap's predictions over a 20-trading-day period. Each daily prediction is performed using a different hybrid quantum network.

Table 5
Comparison of the proposed quantum approach (QuantumLeap) with the corresponding classical neural network and ARIMA using the RMSV metric.

Stock / Metrics	RMSV - Quantum	RMSV - Classical	RMSV - ARIMA
AAPL	**2.83**	6.05	8.37
AMZN	**37.86**	146.35	139.70
BA	**2.63**	10.09	4.98
BAC	**0.33**	1.34	0.61
BP	**0.25**	1.11	2.07
CAT	**0.78**	3.59	9.23
CSCO	**1.35**	1.42	2.08
GE	**0.07**	1.48	2.23
GOOGL	**8.64**	45.73	76.85
HD	**1.93**	14.27	6.40
IBM	**0.70**	5.02	2.14
JNJ	**0.50**	1.49	2.99
JPM	**0.81**	3.18	2.48
JWN	**0.28**	0.90	0.73
KO	**0.14**	1.65	1.59
MFC	**0.15**	1.29	0.64
MMM	**0.70**	3.20	2.28
MRK	**0.28**	1.18	1.98
MSFT	**1.39**	8.71	8.93
ORCL	**0.21**	1.49	3.25
PFE	**0.21**	0.32	0.80
RDSB	**0.20**	1.57	1.44
TD	**0.38**	3.57	1.41
WMT	**0.72**	9.74	8.32

Table 6
Comparison of relative error distributions for the proposed quantum approach (QuantumLeap), corresponding classical neural network and ARIMA using a non-parametric $Mann–Whitney–Wilcoxon$ test to estimate $p-value$.

Architecture	p-value
QNN/Classical NN	0.000066
QNN/ARIMA	0.000093
Classical NN/ARIMA	0.94

References

Adcock, J., Allen, E., Day, M., Frick, S., Hinchliff, J., Johnson, M., Morley-Short, S., Pallister, S., Price, A., & Stanisic, S. (2015). Advances in quantum machine learning. arXiv preprint arXiv:1512.02900.

Almasarweh, M., & Alwadi, S. (2018). ARIMA model in predicting banking stock market data. *Modern Applied Science*, *12*(11), 4.

Ariyo, A. A., Adewumi, A. O., & Ayo, C. K. (2014). Stock price prediction using the ARIMA model. In *2014 UKSim-AMSS 16th international conference on computer modelling and simulation* (pp. 106–112). IEEE.

Ashraf, B. N. (2020). Stock markets' reaction to COVID-19: cases or fatalities? *Research in International Business and Finance*, Article 101249.

Baaquie, B. E. (2009). *Interest rates and coupon bonds in quantum finance*. Cambridge University Press.

Beer, K., Bondarenko, D., Farrelly, T., Osborne, T. J., Salzmann, R., Scheiermann, D., & Wolf, R. (2020). Training deep quantum neural networks. *Nature Communications*, *11*(1), 1–6.

Biamonte, J., Wittek, P., Pancotti, N., Rebentrost, P., Wiebe, N., & Lloyd, S. (2017). Quantum machine learning. *Nature*, *549*(7671), 195–202.

Brée, D. S., Challet, D., & Peirano, P. P. (2013). Prediction accuracy and sloppiness of log-periodic functions. *Quantitative Finance*, *13*(2), 275–280.

Chiribella, G., D'Ariano, G. M., & Perinotti, P. (2008). Quantum circuit architecture. *Physical Review Letters*, *101*(6), Article 060401.

Cohen-Tannoudji, C., Diu, B., & Laloe, F. (2005). *Quantum mechanics, Vol. 1, 231*. Singapore: Wiley.

Cornuejols, G., & Tütüncü, R. (2006). *Optimization methods in finance, Vol. 5*. Cambridge University Press.

Egger, D. J., Gambella, C., Marecek, J., McFaddin, S., Mevissen, M., Raymond, R., Simonetto, A., Woerner, S., & Yndurain, E. (2020). Quantum computing for finance: state of the art and future prospects. *IEEE Transactions on Quantum Engineering*.

Egger, D. J., Gutierrez, R. G., Mestre, J. C., & Woerner, S. (2020). Credit risk analysis using quantum computers. *IEEE Transactions on Computers*.

Elliott, C. (2002). Building the quantum network. *New Journal of Physics*, *4*(1), 46.

Elzerman, J., Hanson, R., van Beveren, L. W., Tarucha, S., Vandersypen, L., & Kouwenhoven, L. (2005). Semiconductor few-electron quantum dots as spin qubits. In *Quantum dots: A doorway to nanoscale physics* (pp. 25–95). Springer.

Farhi, E., Goldstone, J., Gutmann, S., & Sipser, M. (2000). Quantum computation by adiabatic evolution. arXiv Preprint quant-ph/0001106.

Farmer, R. E. (2012). The stock market crash of 2008 caused the great recession: Theory and evidence. *Journal of Economic Dynamics and Control*, *36*(5), 693–707.

Haven, E. E. (2002). A discussion on embedding the black–scholes option pricing model in a quantum physics setting. *Physica A: Statistical Mechanics and its Applications*, *304*(3–4), 507–524.

He, P., Sun, Y., Zhang, Y., & Li, T. (2020). COVID–19's impact on stock prices across different sectors—An event study based on the Chinese stock market. *Emerging Markets Finance and Trade*, *56*(10), 2198–2212.

Liu, J., Lim, K. H., Wood, K. L., Huang, W., Guo, C., & Huang, H.-L. (2019). Hybrid quantum-classical convolutional neural networks. arXiv preprint arXiv:1911.02998.

Lloyd, S., Mohseni, M., & Rebentrost, P. (2013). Quantum algorithms for supervised and unsupervised machine learning. arXiv preprint arXiv:1307.0411.

Lloyd, S., Mohseni, M., & Rebentrost, P. (2014). Quantum principal component analysis. *Nature Physics*, *10*(9), 631–633.

Ma, M., & Mao, Z. (2020). Deep-convolution-based LSTM network for remaining useful life prediction. *IEEE Transactions on Industrial Informatics*, *17*(3), 1658–1667.

Masujima, M. (2008). *Path integral quantization and stochastic quantization*. Springer Science & Business Media.

Meyler, A., Kenny, G., & Quinn, T. (1998). Forecasting irish inflation using ARIMA models.

Mugel, S., Kuchkovsky, C., Sanchez, E., Fernandez-Lorenzo, S., Luis-Hita, J., Lizaso, E., & Orus, R. (2020). Dynamic portfolio optimization with real datasets using quantum processors and quantum-inspired tensor networks. arXiv preprint arXiv:2007.00017.

Müller-Hermes, A., & Reeb, D. (2017). Monotonicity of the quantum relative entropy under positive maps. In *Annales henri poincaré, Vol. 18* (pp. 1777–1788). Springer.

Nielsen, M. A., & Chuang, I. (2002). Quantum computation and quantum information.

Orús, R., Mugel, S., & Lizaso, E. (2019). Forecasting financial crashes with quantum computing. *Physical Review A*, *99*(6), Article 060301.

Öztürk, O., & Wolfe, D. A. (2000). An improved ranked set two-sample mann-whitney-wilcoxon test. *Canadian Journal of Statistics*, *28*(1), 123–135.

de Prado, M. L. (2015). Generalized optimal trading trajectories: A financial quantum computing application. Available at SSRN, 2575184.

Rebentrost, P., Gupt, B., & Bromley, T. R. (2018). Quantum computational finance: Monte Carlo pricing of financial derivatives. *Physical Review A*, *98*(2), Article 022321.

Rebentrost, P., Mohseni, M., & Lloyd, S. (2014). Quantum support vector machine for big data classification. *Physical Review Letters*, *113*(13), Article 130503.

Rosenberg, G., Haghnegahdar, P., Goddard, P., Carr, P., Wu, K., & De Prado, M. L. (2016). Solving the optimal trading trajectory problem using a quantum annealer. *IEEE Journal of Selected Topics in Signal Processing*, *10*(6), 1053–1060.

Schuld, M., Sinayskiy, I., & Petruccione, F. (2014). The quest for a quantum neural network. *Quantum Information Processing*, *13*(11), 2567–2586.

Wiebe, N., Kapoor, A., & Svore, K. M. (2014). Quantum deep learning. arXiv preprint arXiv:1412.3489.

Woerner, S., & Egger, D. J. (2019). Quantum risk analysis. *Npj Quantum Information*, *5*(1), 1–8.

Zhang, K., Zhong, G., Dong, J., Wang, S., & Wang, Y. (2019). Stock market prediction based on generative adversarial network. *Procedia Computer Science*, *147*, 400–406.

Zieliński, R. (1990). Robustness of the one-sided Mann—Whitney—Wilcoxon test to dependency between samples. *Statistics & Probability Letters*, *10*(4), 291–295.