Quantum Machine Intelligence (2023) 5:46
https://doi.org/10.1007/s42484-023-00136-x

RESEARCH ARTICLE

The cross-sectional stock return predictions via quantum neural network and tensor network

Nozomu Kobayashi[1] · Yoshiyuki Suimon[1] · Koichi Miyamoto[2] · Kosuke Mitarai[2,3]

Received: 6 June 2023 / Accepted: 9 November 2023 / Published online: 7 December 2023

Abstract
In this paper, we investigate the application of quantum and quantum-inspired machine learning algorithms to stock return predictions. Specifically, we evaluate the performance of quantum neural network, an algorithm suited for noisy intermediate-scale quantum computers, and tensor network, a quantum-inspired machine learning algorithm, against classical models such as linear regression and neural networks. To evaluate their abilities, we construct portfolios based on their predictions and measure investment performances. The empirical study on the Japanese stock market shows the tensor network model achieves superior performance compared to classical benchmark models, including linear and neural network models. Though the quantum neural network model attains the lowered risk-adjusted excess return than the classical neural network models over the whole period, both the quantum neural network and tensor network models have superior performances in the latest market environment, which suggests capability of model's capturing non-linearity between input features.

Keywords Quantum machine learning · Tensor network · Finance · Stock return prediction

1 Introduction

The arrival of real quantum computers and experiments that show the quantum supremacy (Arute et al. 2019; Madsen et al. 2022) make it more realistic that the new computational paradigm will come by virtue of quantum computing. It is true that we are currently at the era of NISQ (noisy intermidiate-scale quantum computer) (Preskill 2018) and must implement the quantum error correction for a full picture of such a new paradigm, but rapid progress of quantum technologies already open a new window to the research in various fields, such as quantum chemistry, optimization, machine learning, and finance. It is therefore worth looking for a practical application of quantum computers even in the NISQ era. The framework of variational quantum algorithms (VQAs) (Cerezo et al. 2021) is thought to be an effective approach towards the goal. It has been applied, for instance, to solve machine learning problems (Mitarai et al. 2018; Schuld et al. 2019).

Machine learning techniques developed within the framework of VQAs are essentially equivalent to the ones using tensor networks (Huggins et al. 2019; Stoudenmire and Schwab 2016; Stoudenmire 2018, which is originally invented as a tool to simulate quantum physics in classical computers (Fannes et al. 1992; White 1992). Its ability to utilize an exponentially large tensor into a factorized series of smaller tensors has also allowed the machine learning community to successfully solve various machine learning problems (Novikov et al. 2016; Stoudenmire and Schwab 2016; Stoudenmire 2018). It can consequently be considered a quantum-inspired machine learning algorithm.

Given these growing interests of quantum and quantum-inspired machine learning algorithms, it is important to study their applicability on the real-world problems, which are, however, less known so far partly due to the current limitation of computational resource for quantum computers and their simulators. In this work, to address the above issue, we consider a real-world financial problem, namely the prediction of stock returns, employing quantum and quantum-inspired machine learning algorithms.

✉ Nozomu Kobayashi
nozomu.kobayashi@nomura.com

[1] Data Science Department, Nomura Securities Co., Ltd., Tokyo, Japan

[2] Center for Quantum Information and Quantum Biology, Osaka University, Osaka, Japan

[3] Graduate School of Engineering Science, Osaka University, Osaka, Japan

Stock return prediction has been a principal problem in finance. Ever since the work by Fama and French (Eugene and French 1992), who have provided the empirical evidence that the notion of so-called factors is effective in return explainability, significant efforts have been made to find unseen factors that have predictable powers for stock returns. Among practical investors, multi-factor models, which is a linear regression of stock returns by a set of factors, are commonly used thanks to their simplicity and interpretability, though they lack expressibility due to the absence of interaction terms between factors. As it is, machine learning has been becoming an alternative to them. Various studies, Abe and Nakayama (2018); Chinco et al. (2019); Dixon and Polson (2020); Gu et al. (2020, 2021) to name but a few, are conducted on stock return predictions with machine learning methods, which can capture non-linearity in contrast to multi-factor models.

Our interest here is to test whether the quantum or quantum-inspired techniques can be applied to predict stock returns and also have a competitive advantage over classical machine learning algorithms in that task. To this end, using a set of stocks in the Japanese stock market, we conduct portfolio backtesting over 10 years based on stock return predictions by quantum neural network, tensor network, standard linear regression, and neural network and compare their performances. As a result, we find that the tensor network model outperforms the other models, while the quantum neural network model is inferior to the neural network model in the whole backtesting period. We also observe that in the latest market environment, the quantum neural network model has the better performance than the neural network model, which might be related to the overfitting problem. This experiment provides the implication that quantum neural network and tensor network may be able to learn non-linear and interaction effects among features, and they have potential to use in return predictions beyond the conventional models.

This paper is organized as follows. In Sect. 2, we explain the definition of our problem, the stock return predictions, and then describe both classical and quantum machine learning algorithms we use in our analysis. Section 3 presents the methodology to conduct our backtesting experiment and then shows its results, using quantitative metrics that are often used to evaluate an investment performance. Finally, in Sect. 4, we conclude our analysis and discuss some future directions for further research.

2 Methodology

This section collects all the ingredients we use in our analysis. First, we set up the definition of our objective as stock return prediction by cross-sectional analysis, which is based on comparing each stock to others at a point in time, and describe general methodology to tackle the problem. Then, we explain classical models for return predictions, namely the linear regression and the neural network models. Both models are used as benchmark models against quantum ones in our experiment, according to the following reasons. The linear regression model is one of the most traditional models as well as widely employed both by academicians and practitioners in finance. The neural network model serves as a classical counterpart of quantum models, not to mention that it shows superior investing performance over the linear model thanks to its flexibility and expressibility. After that, we introduce quantum circuit learning, which is one realization of quantum neural network, and tensor network algorithms in our framework. Finally, we describe the optimization procedure for these machine learning models.

2.1 Problem definition

The objective of this work is to predict stock returns over cross-section. Before formulating our problem, let us clarify some notations.

Suppose there are trading dates indexed as $0 \leq t \leq T$, and at each trading date t, we have N_t stocks available to invest. We call such a whole set of stocks as stock universe and denote U_t. Remark that the frequency of trading periods depends on our purpose and data availability, by which we adjust the frequency for monthly, weekly, daily, and so on. We describe most generic situation that stock universe varies over time. Stocks are indexed as $i = 1, \cdots, N_t$, and price of i-th stock at time t is denoted as $p_{i,t}$. The return of i-th stock from t to $t + 1$ is then calculated as

$$r_{i,t+1} = \frac{p_{i,t+1} - p_{i,t}}{p_{i,t}}, \qquad (1)$$

which is what we wish to predict. For financial practitioners, it is essential to predict stock returns, since they usually build trading strategies based on predicted future returns. In academic literature, it has been a central problem to investigate the predictability of stock returns and to construct prediction models which satisfies empirical characteristics, with the hope to reveal market structures.

There are mainly two distinct approaches to predict stock returns: one is that by focusing on a specific stock, we use time series analysis to predict its return, and the other is that we predict cross-sectional relative stock performance for whole stock universe at each time, employing each firm's features.

In this work, we adopt the latter cross-sectional approach, in which we leverage information of firms. Suppose we have n features for i-th stock at time t. Such features are compiled to n-dimensional vector $X_{i,t}$, by means of which we describe the general formula of our prediction model as follows:

$$r_{i,t+1} = F(X_{i,t}) + \epsilon_{i,t} \,, \tag{2}$$

where the form of F is not specified here and will be determined by our choice of models. $\epsilon_{i,t}$ represents an error term.

As for a choice of features, what should explain stock returns is a long-standing subject to study in financial literature and has industriously been investigated. The celebrated work by Fama and French Eugene and French (1992) proposes and empirically shows that returns of individual firms can be explained by the following three factors: market (how the whole market moves), size (how large the market capitalization of stocks is), and value (how the stock price is overvalued or undervalued). Ever since their publication, successive studies have followed in order to find other unknown factors to explain returns, with the result that the number of proposed factors has surpassed a hundred. Other than the famous three factors mentioned above, typical factors considered so far include momentum (how big the past return of stocks is) and quality (how stable earnings stocks have).

Regarding the prediction model F, linear regression has been traditionally used both for academicians and practitioners, because of its simplicity and interpretability. In this case, Eq. 2 becomes

$$r_{i,t+1} = \sum_{k=1}^{n} X_{i,t,k} \cdot \theta_k + \epsilon_{i,t} \,, \tag{3}$$

where θ is an n-dimensional vector of model parameters. Note that the index k represents k-th element of an n-dimensional vector. In our analysis, we employ this linear regression model as a benchmark. The parameter θ is determined by the usual ordinary least square regression.

Traditional linear regression models neglect interaction terms between features and non-linear terms. Machine learning models shed light on these issues, as is widely reported in, e.g., Abe and Nakayama (2018); Chinco et al. (2019); Dixon and Polson (2020); Gu et al. (2020, 2021). In our analysis, we use the neural network models as classical machine learning ones. As quantum and quantum-inspired machine learning models, we propose to employ quantum circuit learning and tensor network in return predictions. The following subsections are devoted to describing these methods and how they can be applied for stock return predictions.

2.2 Neural network

We consider a feed-forward neural network, which consists of L layers of affine maps and activation functions. It is formally written as

$$F^{\mathrm{NN}} = \mathsf{W}_{\mathsf{L}} \circ \sigma_{L-1} \circ \cdots \sigma_1 \circ \mathsf{W}_1 \,, \tag{4}$$

The affine map W_l acts on n_l-dimensional input vector Z_l as follows:

$$\mathsf{W}_l(Z_l) = W_l Z_l + b_l \,, \tag{5}$$

where $W_l \in \mathbb{R}^{n_{l+1} \times n_l}$ denotes a weight matrix and $b_l \in \mathbb{R}^{n_l}$ a bias vector. The activation function σ_l is a key to generate a non-linear effect on the model. Though there are plenty of possibilities for what activation function to use, in our analysis, we use the same function ReLU for all $l = 1, \cdots L$ defined as follows:

$$\sigma_l(x) = \mathsf{ReLU}(x) \equiv \max\{x, 0\} \,. \tag{6}$$

Having these in our hands, we construct return prediction such that

$$r_{i,t+1} = F^{\mathrm{NN}}(X_{i,t}) + \epsilon_{i,t} \,. \tag{7}$$

2.3 Quantum circuit learning

Among various quantum machine learning algorithms that have been developed recently (Cerezo et al. 2021), we employ the framework called quantum circuit learning (Mitarai et al. 2018) in this work. It is one of the variational quantum algorithms, aiming at application for supervised machine learning problems. Quantum circuit learning can be regarded as a quantum counterpart of the neural network, since both algorithms try to optimize parameters variationally so that an objective function is minimized. For this reason, quantum circuit learning and similar approaches are also sometimes referred to as quantum neural network (Cerezo et al. 2021).

Quantum circuit learning consists of the following procedures. Suppose we have a dataset consisting of input data $\{x_i\}_{i=1}^N$, and corresponding teacher data $\{y_i\}_{i=1}^N$. First, we construct a quantum circuit $V(x)$ from x. We apply it to some initial state $|\psi_0\rangle$ in order to encode the information of input variables into the quantum state: $|\psi_{in}\rangle = V(x)|\psi_0\rangle$. Then, we prepare a parameterized quantum circuit $U(\theta)$ and apply it to the above state: $|\psi_{out}\rangle = U(\theta)|\psi_{in}\rangle$. Finally, we measure the expectation value $\langle\psi_{out}|O|\psi_{out}\rangle$ of some observable O. In this work, we take the Pauli Z operator acting on the first qubit, Z_1, as the observable O. It is taken as an output of the algorithm $F^{\mathrm{QCL}}(x,\theta)$. The objective function built from y_i and $F^{\mathrm{QCL}}(x,\theta) = \langle\psi_{out}|Z_1|\psi_{out}\rangle$ is minimized by varying the parameter θ. With the optimized parameter $\theta = \theta^*$, the trained model is given as $F^{\mathrm{QCL}}(x,\theta^*)$. Figure 1 shows the general circuit of the quantum circuit learning algorithm.

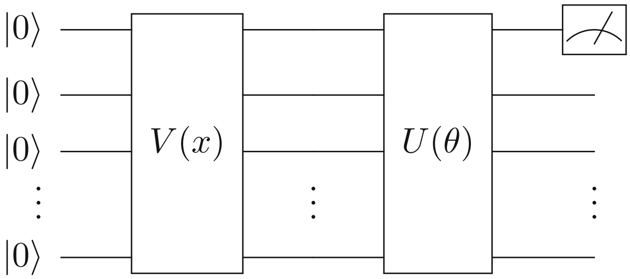

Fig. 1 The general structure of quantum circuit learning, where we have two quantum circuit architectures: data encoding circuit $V(x)$ and parameterized quantum circuit $U(\theta)$

We next explain the construction of quantum circuits for our analysis. It follows Mitarai et al. (2018). Remark that we can choose different forms of encoding and parameterized circuits, which may result in different predicting performance. We do not touch upon the effects of employing different quantum circuits in this paper, leaving it for future investigation. The initial state $|\psi_{in}\rangle$ is prepared as $|0\rangle^{\otimes n}$, where we assume the dimension of vectors x_i of input data is n. The encoding circuit $V(x_i)$ is given by

$$V(x_i) = \prod_{j=1}^{n} R_j^Z(\cos^{-1} x_{i,j}^2) R_j^Y(\sin^{-1} x_{i,j}) \,, \tag{8}$$

where R_j^Z and R_j^Y represent rotation gates acting on j-th qubit:

$$R_j^Z(\phi) = e^{i\phi Z_j/2} \,, \quad R_j^Y(\phi) = e^{i\phi Y_j/2} \,. \tag{9}$$

Note that the input vector x_i must be normalized in the range of $[-1, 1]$.

Then, our parameterized quantum circuit is constructed as follows:

$$U(\theta) = \prod_{i=1}^{d} \left(\prod_{j=1}^{n} U(\theta_j^{(i)}) U_{\text{rand}} \right) \,, \tag{10}$$

which is illustrated in Fig. 2. Here, U_{rand} denotes a time evolution gate for the following Hamiltonian:

$$U_{\text{rand}} = e^{-iH\tau} , \quad H = \sum_{j=1}^{n} a_j X_j + \sum_{j=1}^{n} \sum_{k=1}^{j-1} J_{jk} Z_j Z_k \,, \tag{11}$$

where a_j and $J_{j,k}$ are randomly taken from a uniform distribution on $[-1, 1]$ and τ represents a time length of the evolution. Both of these parameters are fixed during the algorithm. $U(\theta_j^{(i)})$ denotes a sequence of rotation gates on j-th qubit:

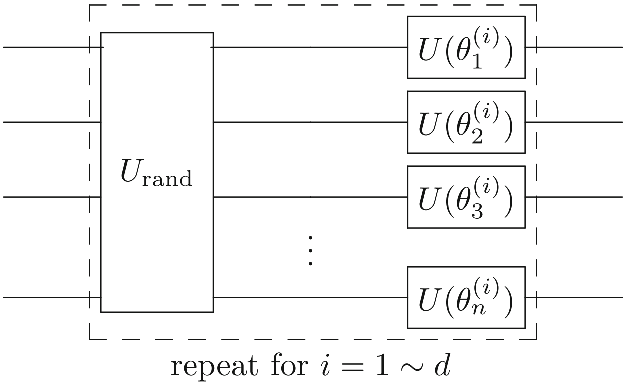

Fig. 2 Our choice of a parameterized quantum circuit in the quantum circuit learning algorithm

$$U(\theta_j^{(i)}) = R_j^X\left(\theta_{j1}^{(i)}\right) R_j^Z\left(\theta_{j2}^{(i)}\right) R_j^X\left(\theta_{j3}^{(i)}\right) \,. \tag{12}$$

where $R_j^X(\phi) = e^{i\phi X_j/2}$. U_{rand} and $U(\theta_j^{(i)})$ are repeatedly applied to the state for d times, resulting in the whole gate $U(\theta)$ in Eq. 10.

Equipped with these gates, quantum circuit learning can be used in return prediction such that

$$r_{i,t+1} = F^{\text{QCL}}(X_{i,t}, \theta) + \epsilon_{i,t} \,. \tag{13}$$

2.4 Tensor network

Tensor network enables us to obtain effective representations of quantum wavefunctions that live on exponentially large dimensional Hilbert space. This is beneficial not only for quantum physics but also for machine learning problems, as tensor network enables us to manipulate a high-dimensional feature space.

The matrix product state (MPS), one of the best-studied and understood tensor networks among all types of ones, is employed in our analysis. MPS is defined as follows. Suppose we have an n-th order tensor T, whose component is given by $T_{i_1\cdots i_n}$. The MPS is a representation of such a tensor T by a product of smaller tensors:

$$T_{i_1\cdots i_n} = \sum_{\alpha_1\cdots\alpha_n} A_{i_1\alpha_1}^{(1)} A_{i_2\alpha_1\alpha_2}^{(2)} \cdots A_{i_n,\alpha_n}^{(n)} \,, \tag{14}$$

where the range of indices α_i is called the bond dimension m.

We follow the approach taken in Novikov et al. (2016); Efthymiou et al. (2019); Stoudenmire (2018) to apply the MPS to our purposes. Consider input vector x and a feature map $\Phi(x)$ which maps x to an n-th order tensor defined as

$$\Phi_{i_1 \cdots i_n}(x) = \phi_{i_1}(x_1)\phi_{i_2}(x_2)\cdots\phi_{i_n}(x_n)\,, \tag{15}$$

where

$$\phi(x_j) = \begin{pmatrix} 1 \\ x_j \end{pmatrix}\,. \tag{16}$$

We construct a model regression function with an MPS W^{MPS} which acts as variational parameters to be trained as

$$y = F^{\text{MPS}}(x, W) = \sum_{i_1 \cdots i_n} W^{\text{MPS}}_{i_1 \cdots i_n} \Phi_{i_1 \cdots i_n}(x)\,. \tag{17}$$

We use this function $F^{\text{MPS}}(W, x)$ in return prediction:

$$r_{i,t+1} = F^{\text{MPS}}(X_{i,t}, W) + \epsilon_{i,t}\,. \tag{18}$$

2.5 Optimization procedure

Now that we have introduced both classical and quantum machine learning models we test in our analysis, let us briefly describe how the training of models is performed. In this subsection, we denote all the prediction models as $F(X_{i,t}, \theta)$ where θ represent parameters for corresponding model, unless otherwise noted. Given true return data $r_{i,t}$ and predicted one $\tilde{r}_{i,t} = F(X_{i,t})$, our objective function E to be minimized is the mean squared error:

$$E = \frac{1}{NT}\sum_{t=1}^{T}\sum_{i=1}^{N}(\tilde{r}_{i,t} - r_{i,t})^2\,. \tag{19}$$

To archive the minimum, we utilize the stochastic gradient descent technique for all models, which is a common prescription in learning of neural networks. In this framework, parameters are subsequently updated such as

$$\theta \leftarrow \theta - \eta \nabla_\theta E\,, \tag{20}$$

where η represents a hyperparameter and the explicit formula for updating parameters depends on the type of optimizers. As for the quantum circuit learning model $F = F^{\text{QCL}}$, the gradient is calculated by the so-called parameter-shift rule (Mitarai et al. 2018; Schuld et al. 2019).

It is worth mentioning that, in tensor network, gradient descent technique is not a standard way to optimize parameters, since a more physics-oriented optimization algorithm called density matrix renormalization group (DMRG) (White 1992) prevails in many physics applications and is also used in machine learning one (Stoudenmire and Schwab 2016). We, however, work with gradient descent in our analysis, as it is simple to implement on high-level API such as TensorFlow (Abadi et al. 2015) and allows us to compare with other models on equal footing. Note that the DMRG approach is thought to be more sophisticated in updating parameters than gradient descent; therefore, it would be interesting to investigate the difference of performances in optimizing tensor network models. See Efthymiou et al. (2019) for more details.

3 Experiment

In this section, we show our empirical study to evaluate how our proposed models perform in return prediction. Our criteria for the evaluation is how profitable our models are, which can be measured by applying models in investment strategies. For this purpose, we adopt an investment strategy based on models' predictions and conduct the backtesting experiment on past historical data. In the following, we explain our dataset and methodology of the investment strategy, then discuss results of backtesting.

3.1 Dataset

Our dataset, or investment universe U_t, is a set of the Japanese stocks that are constituents of TOPIX500 index. TOPIX500 is a Japanese stock market index, consisting of the 500 most liquid stocks with the largest values of market capitalization among members of stocks listed on the Tokyo Stock Exchange.

Input features we use are summarized in Table 1. We consider ten features, which is rather a small number compared to general machine learning models for stock return predictions, where we typically employ as many as tens to hundreds of features to gain expressibility and accuracy. This is due to the fact that our quantum circuit learning architecture requires one qubit for each feature; the more qubits we use, the more computationally intense the simulation of quantum circuits becomes. We therefore limit the number of features to $n = 10$ so that our backtesting experiment can be conducted within reasonable computational time.

As a preprocessing, all features and returns are cross-sectionally ranked at each time step (Gu et al. 2020; Nakagawa et al. 2020): the ith feature of the lth stock at time t, $x_{i,t,l}$, is converted to $(\rho_{i,t,l})/(N_t - 1)$, where $\rho_{i,t,l}$ is the rank of $x_{i,t,l}$ among $\{x_{i,t,l}\}_{i=1,\ldots,N_t}$ in the ascending order.

3.2 Investment strategy

The investment strategy that we take in this work is as follows. Our backtesting period goes from June 2008 to May 2021, during which we make investment decisions on a monthly

Table 1 The list of features and their descriptions

Factor	Feature	Description
Value	Book-value to price ratio	Net asset/market value
	Earning to price ratio	Net profit/Market Value
	Sales to price ratio	Sales/market value
Quality	Return on equity	Net profit/net asset
Momentum	Momentum (1 month)	Stock returns in the last month
	Momentum (3 months)	Stock returns in the past 3 months
	Momentum (6 months)	Stock returns in the past 6months
	Momentum (12 months)	Stock returns in the past 12 months
Size	Market capitalization	log (market value)
Market	Beta	Regression coefficient of stock returns and market return (TOPIX) over 60 months

basis and let t denote the end of each month. Subsequently, our subject to predict is then a 1-month future return.

At the beginning of backtesting, we take 3-year samples (June 2008–May 2011) as training datasets to train the model, and the following 1-year ones (June 2011–May 2012) as test datasets to predict returns. We then roll this procedure forward until the end of the backtesting period. See Fig. 3 for its design. In short, we repeatedly make a prediction for forthcoming year from most recent 3-year samples, but only re-estimate models once a year, not every month, in order to avoid computationally intensive estimation, which is the severe problem for quantum circuit learning running on a simulator.

At each time step t, we sort stocks in descending order based on predicted returns $\tilde{r}_{i,t+1}$ and define a set of stocks belonging to the top quintile as H_t. Assuming our models correctly predict stock returns, H_t should represent most profitable stocks among the whole universe U_t. On that account, we go long, or buy, these stocks with equal weight.

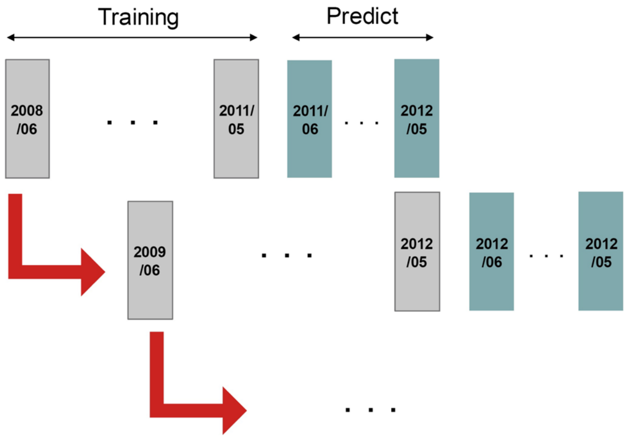

Fig. 3 The concept of our backtesting experiment, showing that we take 3 years as a training period and subsequent 1 year as a test period, rolling this process until the end of the backtesting period

The portfolio return between t and $t+1$ is then given by

$$r_{\text{port},t+1} = \frac{1}{|H_t|} \sum_{i \in H_t} r_{i,t+1} \,, \tag{21}$$

where $|H_t|$ denotes the number of stocks in H_t. We repeat this process and measure the portfolio performance over the backtesting period.

To test the performance of our investment strategy, the common approach is to set up a benchmark portfolio and evaluate excess return between our portfolio and the benchmark. In this work, we use the TOPIX500 index as a benchmark; therefore, the excess return is defined as

$$\alpha_t = r_{\text{port},t} - r_{\text{TOPIX500},t} \,, \tag{22}$$

where $r_{\text{TOPIX500},t}$ denotes the return of the TOPIX500 index at time t. The metrics of the portfolio performance we employ are the following three quantities, all of which are constructed from the time series of α_t:

$$\text{ER} = \prod_{t=1}^{T} (1+\alpha_t)^{12/T} - 1 \,, \tag{23}$$

$$\text{TE} = \sqrt{\frac{12}{T-1} \sum_{t}^{T} (\alpha_t - \bar{\alpha}_t)^2} \,, \tag{24}$$

$$\text{IR} = \text{ER}/\text{TE} \,, \tag{25}$$

with $\bar{\alpha}_t = 1/T \sum_{t=1}^{T} \alpha_t$. Here, ER represents an annualized excess return, TE (tracking error) denotes the corresponding standard deviation, and IR is the so-called information ratio, which expresses the risk-adjusted excess return of the portfolio.

3.3 Model architectures

We summarize the detailed setting of our models. As a traditional model, we use the linear regression which we denote Linear. In all models we consider except for the linear regression, the number of parameters is set to be in the same order for fair comparison. We use Adam optimizer in the training, where the number of epochs is also fixed to 20 in all machine learning models.

Neural network

We prepare two distinct neural network models, which differ in the number of hidden layers.

- NN1 denotes the neural network model with $L = 3$ layers whose nodes are given by $(10, 7, 1)$. This model has 92 parameters to be trained.
- NN2 denotes the neural network model with $L = 4$ layers whose nodes are given by $(10, 5, 4, 1)$. This model has 93 parameters to be trained.

As mentioned earlier, we stick to use ReLU function as the activation function. TensorFlow (Abadi et al. 2015) is used to implement the model.

Quantum circuit learning

We denote our quantum circuit learning model by QCL. The number of qubits is 10, which is the same as the number of input features. The depth of parameterized gates is set to $d = 3$. The number of parameters is consequently 90. We use a quantum circuit simulator Qulacs (Suzuki et al. 2020) to implement and simulate quantum circuits. We have conducted the numerical experiments in a noiseless setting.

Tensor network

We denote our tensor network model by TN. We set the bond dimension to $m = 2$. The number of parameters is then 76 in this setting. We use TensorNetwork (Roberts et al. 2019) as well as TensorFlow for its implementation.

3.4 Backtesting result

Table 2 summarizes the results of our empirical backtesting. See also Fig. 4 for cumulative returns of portfolios and Fig. 5 for cumulative excess returns. We observe that the tensor network model TN has the best performance in regard to both the excess return and the information ratio. On the other hand, the quantum circuit learning model QCL has competitive performance with the neural network model with respect to the excess return; however, it has a larger value of TE, which in turn results in inferior risk-adjusted return IR.

Table 2 The empirical result of backtesting in TOPIX500 universe (Bold characters show the best numbers in each metrics)

	Linear	NN1	NN2	QCL	TN
ER (%)	−0.28	1.27	1.76	1.35	**3.71**
TE (%)	6.64	**3.79**	4.28	6.18	5.41
IR	−0.04	0.34	0.41	0.22	**0.69**

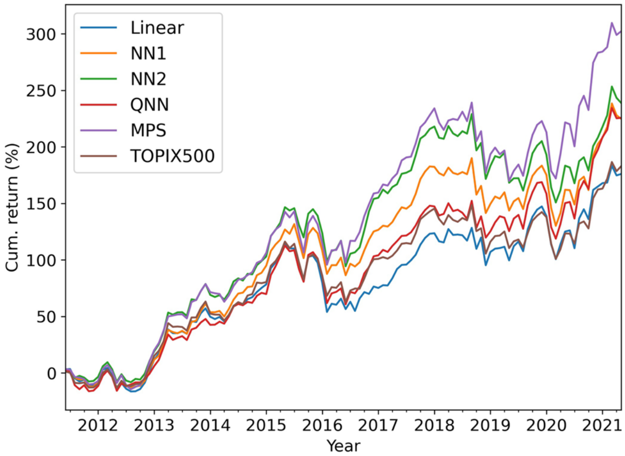

Fig. 4 The cumulative returns of portfolios constructed by various methods and that of TOPIX500

From Fig. 5, before 2016, QCL has the approximately same performance as Linear. This implies that QCL at least learns the linear relationship between input features as is expected. After 2016, on the other hand, QCL continues to outperform Linear, which might be because QCL is able to learn nonlinear relationships as well. What is more, in these recent market environments, QCL can successfully predict stock returns and gain the excess returns, beating classical models. See Appendix for numerics and graphs. We also find that during the last 3 years in the backtesting period, neural network models perform poorly. It suggests that neural

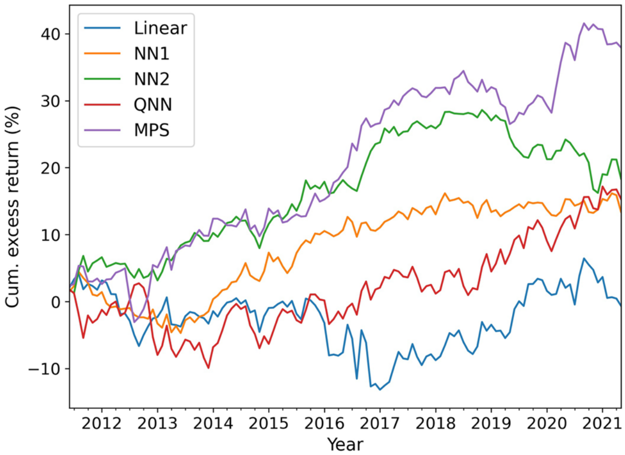

Fig. 5 The cumulative excess returns of portfolio constructed by various methods over TOPIX500

networks used in this analysis tend to overfit to the previous market environment and fail in adapting to the latest one.

The tensor network model TN has the best performance over other models in spite of the lowest number of parameters. It illustrates that TN can possibly have effective architectures to learn financial data, to say nothing of possibility to capture non-linearity among features. It should be further investigated in the future whether this superiority holds when we increase the number of features and parameters in models.

4 Conclusion and discussion

In this paper, we propose to use quantum and its inspired algorithms to predict stock returns. We especially test quantum circuit learning and tensor network as the proposed model against the classical models, namely linear regression and neural network. In order to evaluate their capabilities, we consider the investment strategy based on predicted returns by classical and quantum models. We then conduct backtesting over 10 years in the Japanese stock market.

Our finding is that the tensor network model outperforms classical models, while the quantum circuit learning model archives comparable performance with the neural network models but with higher risk. As is expected, both proposed models seem to learn non-linear relationships between input features, implied by their superior performance against linear regression. Although the performance of the neural network models is deteriorated in the latest years, our proposed models successfully continue to gain the excess return. These differences in the performance can be related to the overfitting problem in machine learning and market instability in these periods. We therefore speculate that quantum techniques can have a good control of the overfitting problem, which is originally suggested in Mitarai et al. (2018). It is, however, unclear whether the hypothesis is true; further examination on this issue should be conducted.

Lastly, we comment on several open problems for future exploration.

- In this work, we evaluate models' capabilities in the Japanese stock market. It should be examined if quantum models work in other countries, e.g., the USA, or in the global market. Nakagawa et al. (2020) studies the transfer learning of neural network in the investment problem between various markets. Whether transfer learning in the quantum model is also effective or not is another interesting research direction.
- While we study the predictability of stocks, it would be interesting whether quantum machine learning is applicable for other assets, such as bonds or currencies. See Suimon et al. (2020); Poh et al. (2022) for the machine learning approach in these assets.
- As is explained in Sect. 2, there are two approaches towards the return prediction, one of which is the cross-section prediction we employ. The other way, namely the time series approach, can be applied in quantum machine learning. In classical neural networks, recurrent neural network and its variants are developed and widely investigated in financial literature (Bao et al. 2017; Kim 2019; Lim et al. 2019; Duan and Kashima 2021). It would be interesting to apply the quantum counterpart of such recurrent networks in financial analysis. See Takaki et al. (2021); Bausch (2020) for the existing literature of quantum recurrent neural networks.

Appendix: Backtesting result from 2016

In this Appendix, we show the backtesting results only from 2016 in Table 3 and Figs. 6 and 7.

Table 3 The empirical result of backtesting in TOPIX500 universe (bold characters show the best numbers in each metrics)

	Linear	NN1	NN2	QCL	TN
ER (%)	−0.12	0.56	0.18	2.63	**4.20**
TE (%)	7.82	**3.48**	4.31	5.85	5.14
IR	−0.02	0.16	0.04	0.45	**0.82**

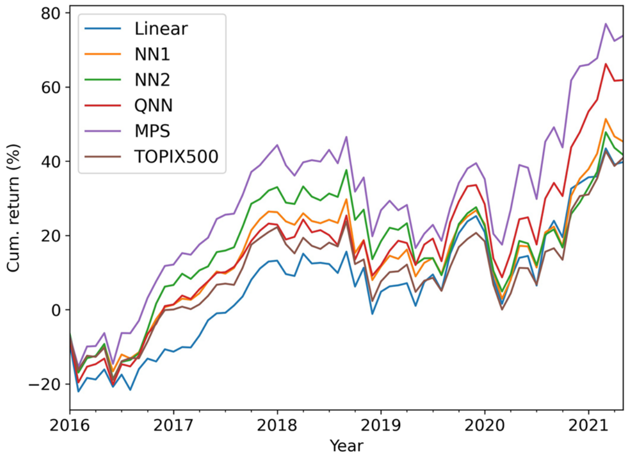

Fig. 6 The cumulative returns of portfolios constructed by various methods from 2016 and that of TOPIX500

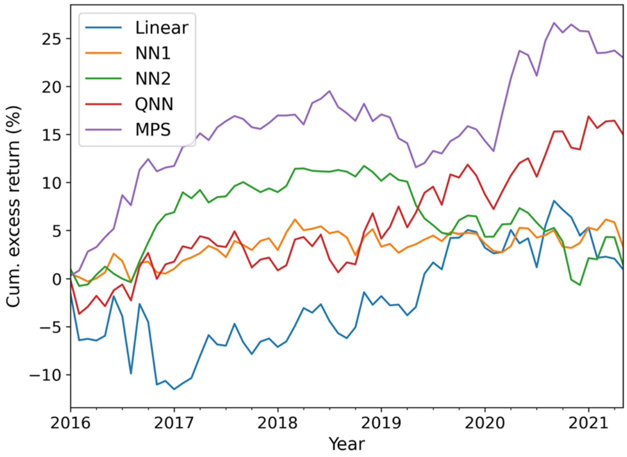

Fig. 7 The cumulative excess returns of portfolios constructed by various methods over TOPIX500 from 2016

Author contribution All authors contributed to the study conception and design. Data collection and analysis were performed by N.K. The first draft of the manuscript was written by N.K., and all authors commented on previous versions of the manuscript. All authors read and approved the final manuscript.

Funding This work was partially supported by KAKENHI grant number JP22K11924 from JSPS, MEXT Q-LEAP grant no. JPMXS0120319794, and JST COI-NEXT no. JPMJPF2014.

Data availability The datasets analyzed during the current study are not publicly available due to the licensing agreements with the data providers.

Declarations

Competing interests The authors declare no competing interests.

References

Abadi M, Agarwal A, Barham P et al (2015) TensorFlow: large-scale machine learning on heterogeneous systems. https://www.tensorflow.org/, software available from tensorflow.org

Abe M, Nakayama H (2018) Deep learning for forecasting stock returns in the cross-section. In: Pacific-Asia conference on knowledge discovery and data mining, Springer, pp 273–284

Arute F, Arya K, Babbush R et al (2019) Quantum supremacy using a programmable superconducting processor. Nature 574(7779):505–510

Bao W, Yue J, Rao Y (2017) A deep learning framework for financial time series using stacked autoencoders and long-short term memory. PloS one 12(7):e0180944

Bausch J (2020) Recurrent quantum neural networks. Advan Neural Inf Process Syst 33:1368–1379

Cerezo M, Arrasmith A, Babbush R et al (2021) Variational quantum algorithms. Nature Rev Phys 3(9):625–644

Chinco A, Clark-Joseph AD, Ye M (2019) Sparse signals in the cross-section of returns. J Finance 74(1):449–492

Dixon M, Polson N (2020) Deep fundamental factor models. SIAM Financial Math 11(3):SC26–SC37

Duan J, Kashima H (2021) Learning to rank for multi-step ahead time-series forecasting. IEEE Access 9:49372–49386

Efthymiou S, Hidary J, Leichenauer S (2019) TensorNetwork for machine learning. arXiv:1906.06329

Eugene F, French K (1992) The cross-section of expected stock returns. J Finance 47(2):427–465

Fannes M, Nachtergaele B, Werner RF (1992) Finitely correlated states on quantum spin chains. Commun Math Phys 144(3):443–490

Gu S, Kelly B, Xiu D (2020) Empirical asset pricing via machine learning. Rev Financial Stud 33(5):2223–2273. https://doi.org/10.1093/rfs/hhaa009. https://academic.oup.com/rfs/articlepdf/33/5/2223/33209812/hhaa009.pdf

Gu S, Kelly B, Xiu D (2021) Autoencoder asset pricing models. J Econ 222(1):429–450

Huggins W, Patil P, Mitchell B et al (2019) Towards quantum machine learning with tensor networks. Quant Sci Technol 4(2):024001

Kim S (2019) Enhancing the momentum strategy through deep regression. Quantitative Finance 19(7):1121–1133

Lim B, Zohren S, Roberts S (2019) Enhancing time-series momentum strategies using deep neural networks. J Financial Data Sci 1(4):19–38

Madsen LS, Laudenbach F, Askarani MF et al (2022) Quantum computational advantage with a programmable photonic processor. Nature 606(7912):75–81

Mitarai K, Negoro M, Kitagawa M et al (2018) Quantum circuit learning. Phys Rev A 98(3):032309

Nakagawa K, Abe M, Komiyama J (2020) RIC-NN: a robust transferable deep learning framework for cross-sectional investment strategy. In: 2020 IEEE 7th international conference on data science and advanced analytics (DSAA), IEEE, pp 370–379

Novikov A, Trofimov M, Oseledets I (2016) Exponential machines. arXiv:1605.03795

Poh D, Lim B, Zohren S et al (2022) Enhancing cross-sectional currency strategies by context-aware learning to rank with self-attention. J Financial Data Sci 4(3):89–107. https://doi.org/10.3905/jfds.2022.1.099. https://jfds.pm-research.com/content/4/3/89.full.pdf

Preskill J (2018) Quantum computing in the NISQ era and beyond. Quantum 2:79. https://doi.org/10.22331/q-2018-08-06-79

Roberts C, Milsted A, Ganahl M et al (2019) TensorNetwork: a library for physics and machine learning. arXiv:1905.01330

Schuld M, Bergholm V, Gogolin C et al (2019) Evaluating analytic gradients on quantum hardware. Phys Rev A 99(3):032331

Stoudenmire E, Schwab DJ (2016) Supervised learning with tensor networks. Advan Neural Inf Process Syst 29

Stoudenmire EM (2018) Learning relevant features of data with multi-scale tensor networks. Quant Sci Technol 3(3):034003

Suimon Y, Sakaji H, Izumi K et al (2020) Japanese interest rate forecast considering the linkage of global markets using machine learning methods. Int J Smart Comput Artif Intell 4(1):1–17. https://doi.org/10.52731/ijscai.v4.i1.500

Suzuki Y, Kawase Y, Masumura Y et al (2020) Qulacs: a fast and versatile quantum circuit simulator for research purpose. arXiv:2011.13524

Takaki Y, Mitarai K, Negoro M et al (2021) Learning temporal data with a variational quantum recurrent neural network. Phys Rev A 103(5):052414

White SR (1992) Density matrix formulation for quantum renormalization groups. Phys Rev Lett 69(19):2863

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.