Expert Systems With Applications 165 (2021) 113973

Contents lists available at ScienceDirect

Expert Systems With Applications

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

Portfolio optimization with return prediction using deep learning and machine learning

Yilin Ma, Ruizhu Han *, Weizhong Wang

School of Economics and Management, Southeast University, 2 Southeast University Road, Jiangning District, Nanjing, 211189, China

ARTICLE INFO

Keywords:
Financial trading
Return prediction
Portfolio optimization
Deep learning
Machine learning

ABSTRACT

Integrating return prediction of traditional time series models in portfolio formation can improve the performance of original portfolio optimization model. Since machine learning and deep learning models have shown overwhelming superiority than time series models, this paper combines return prediction in portfolio formation with two machine learning models, i.e., random forest (RF) and support vector regression (SVR), and three deep learning models, i.e., LSTM neural network, deep multilayer perceptron (DMLP) and convolutional neural network. To be specific, this paper first applies these prediction models for stock preselection before portfolio formation. Then, this paper incorporates their predictive results in advancing mean–variance (MV) and omega portfolio optimization models. In order to present the superiority of these models, portfolio models with autoregressive integrated moving average's return prediction are used as benchmarks. Evaluation is based on historical data of 9 years from 2007 to 2015 of component stocks of China securities 100 index. Experimental results show that MV and omega models with RF return prediction, i.e., RF+MVF and RF+OF, outperform the other models. Further, RF+MVF is superior to RF+OF. Due to the high turnover of these two models, this paper discusses their performance after deducting the transaction fee cased by turnover. Experiments present that RF+MVF still performs the best among MVF models and omega model with SVR prediction (SVR+OF) performs the best among OF models. Moreover, RF+MVF performs better than SVR+OF and high turnover erodes nearly half of their total returns especially for RF+OF and RF+MVF. Therefore, this paper recommends investors to build MVF with RF return prediction for daily trading investment.

1. Introduction

Stock market prediction is a challenging problem of time series prediction since stock market is essentially a nonlinear, dynamic, noisy and chaotic system (Deboeck, 1994). In fact, stock price is influenced by many factors such as political events, company's policies and news, economic situations, interest rates and investors' sentiments (Wang, Wang, Zhang, & Guo, 2011). Recently, many researchers have applied different kinds of machine learning models for stock market prediction and generated satisfying results, such as support vector regression (SVR) (Emir, 2013; Lu, Lee, & Chiu, 2009; Matías & Reboredo, 2012; Rasel, Sultana, & Meesad, 2015) and random forecast (RF) (Ballings, Poel, Hespeels, & Gryp, 2015; Patel, Shah, & Thakkar, 2015). Artificial neural networks (ANNs) (Chong, Han, & Park, 2017; Fischer & Krauss, 2018; Krauss, Do, & Huck, 2017; Oliveira, Cortez, & Areal, 2017; Pang, Zhou, Wang, Lin, & Chang, 2018; Sezer & Ozbayoglu, 2018; Singh & Srivastava, 2017; Zhang & Wu, 2009) as the core of deep learning technology have also been widely used for stock market prediction. Among all the deep learning technologies, LSTM neural network, deep multilayer perceptron (DMLP) and convolutional neural network (CNN) are frequently used in financial time series forecasting (Sezer, Gudelek, & Ozbayoglu, 2020).

Within the superiority of machine learning and deep learning models in stock market prediction, many researchers apply these models in stock preselection process before portfolio formation and generate satisfying results (Huang, 2012; Krauss et al., 2017; Paiva, Cardoso, Hanaoka, & Duarte, 2019; Ta, Liu, & Addis, 2018; Ta, Liu, & Tadesse, 2020; Wang, Li, Zhang, & Liu, 2020). Actually, high quality stock preselection is crucial for the success of portfolio management (Wang et al., 2020). In stock market, individual investors usually try to determine the future return of their investing stocks and then figure out optimal weight for each stock to build a portfolio (Zhang, Li, & Guo, 2018). Thus, after stock preselection process, investors also need to calculate the optimal investment weight for each selected stock before conducting trading investment. This procedure is mainly based on modern portfolio theory (Chen, Zhong, & Chen, 2020; Paiva et al., 2019; Wang et al., 2020). Modern portfolio theory contains different models for

* Corresponding author.
E-mail addresses: ylmaseu@163.com (Y. Ma), daodao-777@163.com (R. Han), wangweizhongky@163.com (W. Wang).

https://doi.org/10.1016/j.eswa.2020.113973
Received 4 June 2020; Received in revised form 1 September 2020; Accepted 3 September 2020
Available online 5 September 2020
0957-4174/

calculating the optimal portfolio weight of each asset. Under given assets, portfolio models are used to optimize one or more objective functions under different constraint conditions. By solving the portfolio optimization problem, the optimal investment weight of each asset is obtained.

Markowitz mean–variance (MV) model as the beginning of modern portfolio theory, builds a portfolio optimization model by simultaneously maximizing portfolio's expected return and minimizing portfolio's investment risk (Fabozzi, Gupta, & Markowitz, 2002). This model forms an efficient frontier, which presents the asset portfolio that decreases the total risk under a predetermined expected return. For each level of expected return, the efficiency frontier gives the optimal investment strategy (Deng, Lin, & Lo, 2012). However, MV model has many limitations for practical application, such as restrict hypotheses and computational complexity for larger scale assets. Thus, numerous models are proposed to solve these issues. For instance, Konno and Yamazaki (1991) develop mean–absolute deviation (MAD) model, where they use absolute deviation to replace variance as risk metric. Alexander and Baptista (2002) propose a mean value at risk (VaR) model by combining MV model with VaR model, in which VaR estimates the tolerated loss under a given confidence level (Jorion, 1997). Since the VaR metric generates many local minima and dissatisfies sub-additive property in managing portfolio risk, Rockafellar and Uryasev (2000) propose conditional value at risk (CVaR) model to overcome the limitations of VaR metric. Kapsos, Christofides, and Rustem (2014) use omega model for portfolio optimization. This model tries to optimize the relative probability of portfolio return or loss exceeding a critical value based on the asymmetric distribution of return by maximizing Omega ratio, since it avoids the limitation of Sharpe ratio.

Classic portfolio optimization models often use mean historical return as expected return, which induces a low pass filtering influence on the stock market's behavior, thus obtaining inaccurate estimates of future short term returns (Freitas, Souza, & Almeida, 2009). Also, since short term stock price is greatly affected by investor sentiment, it is not reasonable to use mean historical return as short term expected return of individual stock. Therefore, stock return prediction should be combined with portfolio optimization models in financial investment (Kolm, Tütüncü, & Fabozzi, 2014). In this regard, many scholars apply predicted return as expected return in building portfolio optimization models (Freitas et al., 2009; Hao, Wang, & Xu, 2013; Zhu, 2013). Not only that, some researches try to combine more predictive results in forming objective functions in portfolio optimization models and further improve the performance of original portfolio optimization models (Ustun & Kasimbeyli, 2012; Yu, Chiou, Lee, & Lin, 2020). To be specific, Ustun and Kasimbeyli (2012) propose a generalized frame of combining stock forecasts in building portfolio optimization model. They develop a portfolio optimization model with eleven objective functions based on mean–variance–skewness model. Subsequently, Yu et al. (2020) adopt this frame by combining return forecasts of autoregressive integrated moving average (ARIMA) in advancing mean–variance (MV) model, mean absolute deviation (MAD) model, downside risk (DSR) model, linearized value-at-risk (LVaR) model, conditional value-at-risk (CVaR) model and omega model. Experimental results show that these advanced portfolio optimization models can further improve the performance of original portfolio optimization models, and extended MV and omega models perform better than other models. Since ARIMA model is mainly based on the hypotheses of linearity and normal distribution, these assumptions may not be satisfied in stock return series. Machine learning models without these restrict hypotheses have shown better performance than ARIMA model (Adebiyi, Adewumi, & Ayo, 2014; Hansen & Nelson, 2002). Also, deep learning models as novel machine learning technologies have shown promising performance in stock market prediction (Hiransha, Gopalakrishnan, Menon, & Soman, 2018; Long, Lu, & Cui, 2018; Moews, Herrmann, & Ibikunle, 2019). Thus, it is interesting and necessary to investigate the combination of machine learning and deep learning models' return prediction in advancing classic portfolio optimization models. As far as we know, there is no existing research focusing on this problem. Also, since RF, SVR, DMLP, LSTM neural network and CNN are frequently used and generate satisfying performance in stock prediction, this paper pays attention to extend portfolio optimization models with these models' predictive results according to the frame in Yu et al. (2020), which can give full play to their advantages in return prediction.

The main purpose of this paper is to research the performance of advancing portfolio optimization models with machine learning and deep learning models' return prediction. In this regard, this study has two contributions to fill the gaps in existing researches. First, this paper researches the performance of two machine learning models (i.e., SVR and RF) and three deep learning models (i.e., DMLP, LSTM neural network and CNN) in the stock preselection process before portfolio formation. These models own overwhelming performance than traditional time series models, which guarantees high quality stocks are selected before building portfolio optimization models. Also, these models need few hypotheses that is more suitable for practical applications. Second, this paper combines the predictive results of these models in advancing classic MV and omega portfolio optimization models for the first time. These advanced portfolio optimization models not only own the advantages of machine learning and deep learning models in return prediction, but also retain the essences of classical MV and omega models in portfolio optimization. Thus, these models can further improve the out-of-sample performance of existing models. In addition, this study applies China Securities 100 Index component stocks as the entire assets. Moreover, this study focuses on the data from 2007 to 2015 and applies the last four years to measure the effect of the proposed investment models.

The remainder of this paper is presented as follows. Section 2 reviews some related works concerning to the this paper. Section 3 describes some related models. Section 4 presents the detailed experimental process. Section 5 shows experimental results. Finally, Section 6 draws a conclusion.

2. Literature review

There are many works concerning stock trading investment by using different kinds of models. This paper only presents some relative researches.

Huang (2012) developed a model for stock selection by using SVR and genetic algorithms (GAs). This model applied SVR to predict future return of each stock where GA was employed to optimize model parameters and input features. Then, top-ranked stocks were equal weighted to build a portfolio. Empirical results showed that the investment performance of proposed model performed better than the benchmarks. Krauss et al. (2017) analyzed the effectiveness of DNN, gradient-boosted trees, RF, and several ensembles of these methods in the context of statistical arbitrage. Each model was trained on lagged returns of all stocks of the $S\&P$ 500 after elimination of survivor bias. Experiments showed that the simple equal-weighted ensemble method could generate satisfying result. Lee and Yoo (2018) compared three kinds of recurrent neural network for stock return prediction, i.e., recurrent neural network, gated recurrent unit and long short-term memory (LSTM) neural network. The experimental results presented LSTM neural network performed the best of these models. Also, they built predictive threshold-based portfolios based on the predictive results of LSTM neural network. This model was more data-driven than existing models in designing portfolio. Experimental results showed that this portfolio earned promising return. Fischer and Krauss (2018) deployed LSTM neural networks for predicting directional movements of the constituent stocks of the S&P 500 from 1992 until 2015. They found that LSTM neural networks based portfolio outperformed memory-free classification based portfolio models, i.e., RF, DNN and LR.

The common shortcoming of above models is that these models only apply simple method to build their portfolio, such as equal weighted and threshold based method. These portfolio construction methods do not analyze the risk of each stock, which unbalances the portfolio's expected return and risk.

Lin, Huang, Gen, and Tzeng (2006) proposed a dynamic portfolio optimization model. This model used Elman neural network to predict future stock return, and then applied covariance matrix to measure risk. Experimental results presented that this model outperformed vector autoregression model in dynamic portfolio selection models. Alizadeh, Rada, Jolai, and Fotoohi (2011) developed a portfolio optimization model by using adaptive neuro-fuzzy inference system for portfolio return prediction and variance index for risk assessment. Experimental results showed that this portfolio optimization model performed better than MV model, neural network and Sugeno–Yasukawa method. This research informed us that combining artificial intelligence technique with modern portfolio optimization could generate better performance than each single model for trading investment. Deng and Min (2013) applied linear regression model with ten variables to select stocks from US and global equities, and built portfolio based on MV model with some practical constraints of risk tolerance, tracking error and turnover. They found that the risk adjusted return of the proposed model of global equity universe performed better than the domestic equity universe, and the portfolio return increased with the systematic tracking error and risk tolerance. Paiva et al. (2019) proposed a unique decision-making model for day trading investments on stock market, which was developed using a fusion approach of SVM and MV model for portfolio selection. The proposed model was compared with two other models, i.e., SVM+ 1/N and Random+ MV. The experimental evaluation was based on assets from Ibovespa stock market, which showed the proposed model performed the best. Wang et al. (2020) developed a portfolio model by using LSTM neural network for stock selection and MV for portfolio optimization. In this model, LSTM neural network first selected k stocks from total stock set, then the chosen k stocks are used to build MV portfolio model. They compared LSTM neural network with SVM, RF and ARIMA model in stock selection process, then used MV for portfolio optimization. Experiments' result showed that their proposed model outperformed the others. Ta et al. (2020) built portfolios by using LSTM neural network and three portfolio optimization techniques, i.e., equal weighted method, Monte Carlo simulation and MV model. Also, they applied linear regression and SVM as comparisons in stock selection process. Experimental results showed that LSTM neural network owned higher predictive accuracy than linear regression and SVM, and its constructed portfolios outperformed the others.

These models apply different methods for stock selection, then build portfolio optimization models with selected stocks for trading investment. These methods show us a promising direction to build portfolio models in practice. However, classic portfolio optimization models are often inappropriate for short term practical investment. Thus, it is important to explore more efficient approach to combine return predictive results with portfolio optimization models.

Freitas et al. (2009) proposed a prediction-based portfolio optimization model by using autoregressive moving reference neural network (AR-MRNN). This model first used AR-MRNN to predict future stock return, then built portfolio optimization model using predictive results of AR-MRNN. Experimental results showed that this model outperformed original portfolio optimization model and beat the market index. Then, Hao et al. (2013) developed a similar portfolio optimization model by using SVR. They compared their model with the model in Freitas et al. (2009). Experimental results showed that their model owned better performance in trading investment. Ustun and Kasimbeyli (2012) built a generalized approach for building portfolio optimization model using stock return prediction. They built an extended mean–variance–skewness model by using predictive returns and predictive return errors to form eleven objective functions, which comprehensively used predictive results in building portfolio optimization models. Following on this approach, Yu et al. (2020) combined ARIMA model's forecasts in advancing six portfolio optimization models (i.e., MV, MAD, DSR, LVaR, CVaR and omega models). They first used ARIMA model to predict future stock return, then applied the predictive results in extending these portfolio optimization models. Experimental results showed that the advanced portfolio optimization models with ARIMA prediction outperformed single portfolio models, and extended MV and omega models with ARIMA prediction performed the best among these models.

These models show us a promising direction to combine stock return prediction with portfolio optimization models, which fully applies the advantages of stock forecasts in building portfolio optimization models. Therefore, this paper tries to follow this step to further research the performance of combining machine learning and deep learning models with portfolio optimization models.

Table 1
Parameters of DMLP.

Parameter	Value
Hidden nodes	5,10,15,20,25,30
Hidden layers	1,2,3, …,10
Learning rate	0.0001, 0.001, 0.01, 0.1
Patient	0,5,10
Batch size	50,100,200
Loss function	Mean absolute error
Optimizer	SGD, RMSprop, Adam

3. Models

This section first introduces the DMLP, LSTM neural network, CNN, SVM and RF models, then their applied parameters are displayed. Also, the classical MV model and omega model are clarified.

3.1. Deep multilayer perceptron (DMLP)

DMLP is a classic ANN, which is different from multilayer perceptron (MLP) since it contains more hidden layers. Although in terms of mapping abilities, MLP is believed to be capable of approximating arbitrary functions (Principe, Euliano, & Lefebvre, 1999), DMLP usually performs better than MLP with few hidden layers in practice (Orimoloye, Sung, Ma, & Johnson, 2020; Singh & Srivastava, 2017). DMLP model contains three parts, i.e., input layer, hidden layer and output layer. In this paper, stochastic gradient descent is used to train DMLP and earlystopping technology is applied to avoid overfitting problem during the training process. The main hyperparameters of DMLP contain hidden nodes, hidden layers, optimizer, learning rate, activation function, loss function, batch size and patient. As recommended by Orimoloye et al. (2020), relu function is adopted as activation function. The considered values of the other hyperparameters are presented in Table 1. Grid research is used to discover the optimal hyperparameter.

After many attempts, the specified topology of DMLP model is discovered. DMLP with 15 nodes each hidden layer, 6 hidden layers, 0.01 as learning rate, 0 as patient, 100 as batch size and Adam as optimizer performs the best. Thus, this paper uses this DMLP model for stock return prediction.

3.2. Long short term memory (LSTM) neural network

LSTM neural network is a kind of recurrent neural network, which was proposed to overcome the limitation of recurrent neural network and retain long term information (Graves & Schmidhuber, 2005). This property is mainly based on the memory cells in hidden layer. LSTM neural network usually consists of input layer, hidden layer and output layer. This paper uses stochastic gradient descent to train LSTM neural

Table 2
Parameters of LSTM neural network.

Parameter	Value
Hidden nodes	5,10,15,20,25,30
Hidden layers	1,2,3, …,10
Learning rate	0.0001,0.001,0.01,0.1
Patient	0,5,10
Batch size	50,100,200
Dropout rate	0.1, 0.2, …,0.5
Recurrent dropout rate	0.1, 0.2, …,0.5
Loss function	Mean absolute error
Optimizer	SGD, RMSprop, Adam

Table 3
Parameters of CNN.

Parameter	Value
Filter numbers	2,4,8,16,32,64
Convolutional layers	1,2,3, …,10
Maxpooling layers	1,2,3, …,10
Fully connected layers	1,2,3,4,5
Fully connected layer nodes	2,4,8,16,32,64
Learning rate	0.0001,0.001,0.01
Patient	0,5,10
Batch size	50,100,200
Activation function	relu, tanh
Loss function	Mean absolute error
Optimizer	SGD, RMSprop, Adam

network and earlystopping technology is applied to reduce overfitting. The considered hyperparameters of LSTM neural network contain hidden nodes, hidden layers, learning rate, batch size, patient, dropout rate, recurrent dropout rate, activation function, optimizer and loss function. According to Orimoloye et al. (2020), relu function is used as activation function. The investigated values of the other hyperparameters are presented in Table 2. Grid research is used to determine the optimal hyperparameter.

By trial and error, the optimal hyperparameters of LSTM neural network are determined. The topology of LSTM neural network consists of 4 hidden layers and each layer contain 5 nodes. And, 0.01, 0, 100, 0.4, 0.3 and RMSprop are set for learning rate, patient, batch size, dropout rate, recurrent dropout rate and optimizer respectively. In the following, this paper uses this LSTM neural network model for return prediction.

3.3. Convolutional neural network (CNN)

CNN is a novel ANN, which was introduced by LeCun and Bengio (1995). CNN is often used in computer vision and image process, and obtains satisfying performance (Ji, Xu, Yang, & Yu, 2012; Long, Shelhamer, & Darrell, 2015). Recent years, some researchers have applied CNN in stock price prediction and generated promising results (Hoseinzade & Haratizadeh, 2019; Sezer & Ozbayoglu, 2018). CNN typically consists of many successive convolutional layers and pooling layers, then followed by some fully connected layers. Since stock return is time series, this paper applies one dimensional (1D) CNN for stock return prediction. Similarly, stochastic gradient descent method is used to train CNN and earlystopping technology is used to avoid the overfitting problem in this paper. The hyperparameters of CNN in this paper contain filter numbers, convolutional layers, maxpooling layers, fully connected layers, fully connected layer nodes, learning rate, patient, batch size, activation function, optimizer and loss function. Their considered values are presented in Table 3. Grid research is applied to discover the optimal hyperparameter.

After multiple trial and error, the topology of CNN is determined. The input layer is followed with one 1D convolutional layer (2 filters with 2 × 1 size), one 1D maxpooling layer (2 × 1 size), three fully connected layers (2 nodes) and output layer. And, learning rate, patient, batch size, activation function and optimizer are assigned as $0.001, 0, 100$, relu and SGD respectively. This paper applies this CNN model for stock return prediction in the following.

Table 4
Parameters of RF.

Parameter	Value
Max-depth	5,10,15,20,25,30
Min-samples-split	2,5,10,15,20,25,30
Min-samples-leaf	1,5,10,15,20,25,30
Max-features	10,20,30,40,50

Table 5
Parameters of SVR.

Parameter	Value
C	$2^0, 2^1, \dots, 2^5$
γ	$2^{-5}, 2^{-4}, \dots, 2^0$

3.4. Random forest (RF)

RF is a nonparametric and nonlinear model, which was first proposed by Ho (1995). This model avoids the overfitting problem since it always converges (Breiman, 2001). Due to the advantages of RF, it is often used for stock prediction (Ballings et al., 2015; Booth, Gerding, & Mcgroarty, 2014; Qin, 2014). The main parameters of RF are number of decision tree, the maximum depth of the tree (max-depth for short), the minimum number of samples required to split an internal node (min-samples-split for short), the minimum number of samples needed to be at a leaf node (min-samples-leaf for short) and the number of features to consider when looking for the best split (max-features for short). In this paper, the number of decision tree is set to 500 according to Breiman (2001). The considered values of the other parameters are presented in Table 4. Grid research is used to determine the optimal parameters.

After many experiments, this paper sets max-depth, min-samples-split, min-samples-leaf and max-features as 20, 10, 10 and 40. Thus, this paper uses this RF model for stock return prediction.

3.5. Support vector regression (SVR)

SVR is an classical machine learning model which has been widely applied in stock market prediction (Emir, 2013; Lu et al., 2009; Matías & Reboredo, 2012; Rasel et al., 2015). SVR uses Vapnik's Structural Risk Minimization (SRM) principle to resolve different regression problems. SVR originates from statistical learning theory, which is applied for how to regulate generalization and discover the optimal trade off between complexity of model structure and empirical risk.

In this paper, radial basis function is used as the kernel function of SVR, which is given as follows:

$$K(x_i, x_j) = \exp(-\gamma \|x_i - x_j\|^2) \tag{1}$$

where γ is the constant of radial basis function.

The parameters of SVR are composed of the regularization parameter (C) and γ. Table 5 presents the used value of these parameters. Grid research is used to discover their optimal values in each training process.

3.6. Autoregressive integrated moving average (ARIMA) model

ARIMA is a classical statistical model, which is often used in stock prediction. The ARIMA model can be presented as follows:

$$(1 - \sum_{i=1}^{p} \phi_i L^i)(1 - L^i)^d r_t = \delta + (1 - \sum_{i=1}^{q} \theta_i L^i)\varepsilon_t \tag{2}$$

where $p, d, q, L, \phi_i, \theta_i$ and ε_t represent the number of autoregressive terms, the number of difference times, the number of moving average terms, lag operator, autoregressive parameter, moving average parameter and error term respectively (Yu et al., 2020). p, d, q need to be determined before using ARIMA model. Since ARIMA model needs some restrict hypotheses such as stationarity test, this paper set the values of p, d, q before each training process.

In this paper, we uses ARIMA model as benchmark in stock return prediction process. Also, its predictive results are combined in advancing MV and omega portfolio optimization models for further comparisons in trading simulation.

3.7. Mean–variance with forecasting (MVF) model

Markowitz (1959) as the forerunner of modern finance theory introduced mean–variance (MV) model, which presented a mathematical solution to settle the trade-off between expected return maximization and risk minimization. Following the frame proposed by Yu et al. (2020), this paper combines the return predictive results in advancing MV model for building the MVF model.

The MVF model is actually a multi-objective optimization problem. The following equations present the MVF model.

$$\min \quad \sum_{i,j=1}^{n} x_i x_j \sigma_{ij} \tag{3}$$

$$\max \quad \sum_{i=1}^{n} x_i \hat{r}_i \tag{4}$$

$$\max \quad \sum_{i=1}^{n} x_i \bar{\varepsilon}_i \tag{5}$$

Subject to

$$\sum_{i=1}^{n} x_i = 1 \tag{6}$$

$$0 \le x_i \le 1 \quad i = 1, 2, \ldots, n \tag{7}$$

where x_i means the proportion of asset i in portfolio, n is the number of assets in portfolio, σ_{ij} is the covariance of asset i and j, $\hat{r}_i$ denotes the predicted return of asset i and $\bar{\varepsilon}_i$ represents the average predictive errors of asset i over the sample period. This paper sets the sample period as 20 trading days to build the MVF model according to Yu et al. (2020). In other words, $\hat{r}_i$ is predicted return of asset i at time t, and $\bar{\varepsilon}_i$ means the average predictive error of asset i over the past 20 trading days, i.e., time $t, t-1, \ldots, t-19$. The predictive error of asset i at time t equals to $\varepsilon_i = r_i - \hat{r}_i$, where r_i represents the actual return of asset i. Eqs. (4)–(5) mean maximization of expected portfolio return and the sample period's abnormal return respectively.

The equal weighted method is often used to convert the above multiple objective portfolio optimization to a single objective model (Yu et al., 2020). Thus, the MVF model becomes the following form:

$$\min \quad \sum_{i,j=1}^{n} x_i x_j \sigma_{ij} - \sum_{i=1}^{n} x_i \hat{r}_i - \sum_{i=1}^{n} x_i \bar{\varepsilon}_i \tag{8}$$

Subject to

$$\sum_{i=1}^{n} x_i = 1 \tag{9}$$

$$0 \le x_i \le 1 \quad i = 1, 2, \ldots, n \tag{10}$$

3.8. Omega with forecasting (OF) model

Omega ratio was first introduced by Keating and Shadwick (2002). Then, it is widely used to build portfolio since it avoids the known limitations of Sharpe ratio (Gilli, Schumann, di Tollo, & Cabej, 2011; Kane, Bartholomew-Biggs, Cross, & Dewar, 2009; Kapsos et al., 2014). The Omega ratio is defined as follows

$$w = \frac{E(y_i) - \tau}{E[\tau - y_i]^{+}} + 1 \tag{11}$$

where τ denotes the threshold of dividing returns into expected (revenue) and unexpected (loss), and it is often decided by investors. y_i means random return of asset i. Since Omega ratio needs the probability distribution of asset returns, the obtained solution turns into biased and overoptimistic when this probability distribution is imprecise (Kapsos et al., 2014). Thus, Kapsos et al. (2014) introduce worst-case omega ratio (WCOR) to solve this problem, and modify omega model as follows.

$$\max \psi \tag{12}$$

Subject to

$$\delta\Big(\sum_{j=1}^{n} x_j \bar{r}_j^i - \tau\Big) - (1-\delta)\frac{1}{T^i}\sum_{t=1}^{T^i} \eta_t^i \ge \psi \tag{13}$$

$$\eta_t^i \ge -\sum_{j=1}^{n} x_j \bar{r}_j^i + \tau \tag{14}$$

$$\eta_t^i \ge 0 \tag{15}$$

$$\sum_{j=1}^{n} x_j = 1 \tag{16}$$

$$0 \le x_i \le 1 \tag{17}$$

$$t = 1, 2, \ldots, T^i \quad i = 1, 2, \ldots, l \quad j = 1, 2, \ldots, n \tag{18}$$

where x_i means the proportion of asset i in the portfolio, η_t^i is an auxiliary variable used to linearize this portfolio model, δ denotes the risk return preference of this model, T^i means the sample period of the ith distribution, $\bar{r}_j^i$ represents the sample period's mean return of the ith distribution, l is the number of distributions and n is the number of assets in portfolio. This paper sets δ, T^i, τ, l as 0.5, 20, 0 and 1 respectively according to Yu et al. (2020).

Then, we can combine return predictive results with this model and build the OF model similar to MVF model.

$$\max \psi \tag{19}$$

$$\max \sum_{i=1}^{n} x_i \hat{r}_i$$

$$\max \sum_{i=1}^{n} x_i \bar{\varepsilon}_i$$

Subject to

$$0.5\Big(\sum_{i=1}^{n} x_i \bar{r}_i\Big) - \frac{0.5}{20}\sum_{t=1}^{20} \eta_t \ge \psi \tag{20}$$

$$\eta_t \ge -\sum_{i=1}^{n} x_i \bar{r}_i \tag{21}$$

$$\eta_t \ge 0 \tag{22}$$

$$\sum_{i=1}^{n} x_i = 1 \tag{23}$$

$$0 \le x_i \le 1 \tag{24}$$

$$t = 1, 2, \ldots, 20 \quad i = 1, 2, \ldots, n \tag{25}$$

Similarly, we convert the multiobjective optimization model to single objective model (Yu et al., 2020).

$$\min -\psi - \sum_{i=1}^{n} x_i \hat{r}_i - \sum_{i=1}^{n} x_i \bar{\varepsilon}_i \tag{26}$$

Subject to

$$0.5\Big(\sum_{i=1}^{n} x_i \bar{r}_i\Big) - \frac{0.5}{20}\sum_{t=1}^{20} \eta_t \ge \psi \tag{27}$$

$$\eta_t \ge -\sum_{i=1}^{n} x_i \bar{r}_i \tag{28}$$

Table 6
Selected stocks' tickers.

000001	000002	000063	000069	000538	000625
000651	000725	000858	000895	002024	300059
600000	600010	600011	600015	600016	600018
600019	600028	600030	600031	600036	600048
600050	600104	600111	600115	600150	600276
600340	600372	600398	600485	600518	600519
600585	600637	600690	600795	600837	600886
600887	600893	600900	601006	601111	601398
601988					

$$\eta_t \geq 0 \tag{29}$$

$$\sum_{i=1}^{n} x_i = 1 \tag{30}$$

$$0 \leq x_i \leq 1 \tag{31}$$

$$t = 1, 2, \dots, 20 \quad i = 1, 2, \dots, n \tag{32}$$

4. Experimental process

In order to evaluate the proposed methods, this paper utilizes the historical data of the China Securities 100 Index component stocks as experimental data set. The China Securities 100 Index is selected from the sample stocks of the Shanghai and Shenzhen 300 Index in order to comprehensively reflect the overall situation of the most influential large capitalization companies in the Shanghai and Shenzhen stock markets. Experimental data range from January 4, 2007 to December 31, 2015, and after deleting the stocks that are unlisted during this period or halted for a long period of time, the remainder of China Securities 100 index component stocks contains 49 stocks, which are presented in Table 6.

For each stock, the past 60 days' daily returns are used as input features to predict next day's return. For each input feature, since its fluctuation range has apparent difference, we need to process them before training models. For each feature series $\{d_i\}$, d_i is processed as follows

$$d_i = \begin{cases} d_m + 5d_{mm} & if \quad d_i \geq d_m + 5d_{mm}, \\ d_m - 5d_{mm} & if \quad d_i \leq d_m - 5d_{mm}. \end{cases} \tag{33}$$

where d_m and d_{mm} mean the median of series $\{d_i\}$ and series $\{|d_i - d_m|\}$ respectively. Last, each processed input feature is standardized in order to unify fluctuation range before model training

$$\hat{d}_i = \frac{d_i - \mu}{\sigma} \tag{34}$$

where μ and σ represent the mean and standard deviation of series $\{d_i\}$ respectively.

The total data set contains 9 years' data. The experiment is implemented by sliding window, i.e., the first 4 years' data is used as training set, the following 1 year's data is applied as validation set, then the next year's data is used to test models' abilities, so we can use the last four years' data (2012–2015) to test the performance of each model. In the experiment, the DMLP, LSTM neural network and CNN model are implemented based on Keras deep learning package and the SVR and RF are prepared based on Scikit-learn machine learning package.

5. Results

This section first presents the predictive results of different models in stock prediction during the whole test period. In the following, this paper conducts trading simulation to compare the performance of different MVF and OF models for daily trading investment without transaction fee. Lastly, this paper further presents the performance of these models when transaction fee is considered.

Table 7
The predictive performance of different models.

Model		MAE	MSE	H_R	H_{R+}	H_{R-}
DMLP	mean	$2.21 * 10^{-2}$	$1.04 * 10^{-3}$	48.71%	48.89%	48.89%
	SD	$5.77 * 10^{-3}$	$5.35 * 10^{-4}$	$2.48 * 10^{-2}$	$3.20 * 10^{-2}$	$3.18 * 10^{-2}$
LSTM	mean	$2.02 * 10^{-2}$	$8.86 * 10^{-4}$	48.46%	48.64%	47.90%
	SD	$4.01 * 10^{-3}$	$3.20 * 10^{-4}$	$2.11 * 10^{-2}$	$3.43 * 10^{-2}$	$3.07 * 10^{-2}$
CNN	mean	$4.24 * 10^{-2}$	$2.32 * 10^{-2}$	48.39%	48.27%	47.88%
	SD	$4.79 * 10^{-2}$	$7.68 * 10^{-2}$	$2.50 * 10^{-2}$	$6.62 * 10^{-2}$	$1.17 * 10^{-1}$
SVR	mean	$3.10 * 10^{-2}$	$1.70 * 10^{-3}$	47.94%	52.83%	47.69%
	SD	$1.20 * 10^{-2}$	$9.71 * 10^{-4}$	$2.38 * 10^{-2}$	$1.47 * 10^{-1}$	$2.36 * 10^{-2}$
RF	mean	$1.85 * 10^{-2}$	$8.08 * 10^{-4}$	48.69%	48.99%	48.28%
	SD	$3.92 * 10^{-3}$	$3.02 * 10^{-4}$	$2.15 * 10^{-2}$	$3.25 * 10^{-2}$	$2.35 * 10^{-2}$
ARIMA	mean	$2.94 * 10^{-2}$	$2.12 * 10^{-3}$	48.05%	48.33%	47.68%
	SD	$7.00 * 10^{-3}$	$1.77 * 10^{-3}$	$1.96 * 10^{-2}$	$2.95 * 10^{-2}$	$2.40 * 10^{-2}$

SD means standard deviation.

5.1. Stock return prediction

In order to comprehensively measure the performance of different models in stock return prediction process, five metrics, i.e., mean squared error (MSE), mean absolute error (MAE), H_R, H_{R+} and H_{R-}, are applied in this paper. As these metrics clearly show the model predictive ability, they are widely used as performance metrics (Freitas et al., 2009; Gandhmal & Kumar, 2019; Wang et al., 2020). These metrics are defined as follows

$$MSE = \frac{1}{N} \sum_{t=1}^{N} (r_t - \hat{r}_t)^2 \tag{35}$$

$$MAE = \frac{1}{N} \sum_{t=1}^{N} |r_t - \hat{r}_t| \tag{36}$$

$$H_R = \frac{Count_{t=1}^{n}(r_t \hat{r}_t > 0)}{Count_{t=1}^{n}(r_t \hat{r}_t \neq 0)} \tag{37}$$

$$H_{R+} = \frac{Count_{t=1}^{n}(r_t > 0 \quad AND \quad \hat{r}_t > 0)}{Count_{t=1}^{n}(\hat{r}_t > 0)} \tag{38}$$

$$H_{R-} = \frac{Count_{t=1}^{n}(r_t < 0 \quad AND \quad \hat{r}_t < 0)}{Count_{t=1}^{n}(\hat{r}_t < 0)} \tag{39}$$

where $r_t, \hat{r}_t$ represent actual return and predictive return at time t respectively. In addition, H_R denotes total hit rate, H_{R+} means accuracy of positive prediction and H_{R-} is accuracy of negative prediction. Note that this paper sets MAE and MSE as the key metrics since they play important roles in building portfolio with return prediction.

First, the performance of three deep learning models is compared. Table 7 shows that LSTM neural network owns the lowest MAE and MSE among three deep learning models, also its standard deviation is the lowest. Second, for two machine learning models, RF's MAE and MSE are lower than SVR, and its standard deviation is much lower. Third, compared with ARIMA model, RF and LSTM neural network have lower MAE and MSE, also their standard deviations are smaller. Thus, RF and LSTM neural network both perform better than traditional ARIMA model, which is consistent with the conclusion in Adebiyi et al. (2014) and Hiransha et al. (2018). In addition, the predictive errors of RF and LSTM neural network, i.e., the values of MAE and MSE, are also superior than existing literature (Sadaei, Enayatifar, Lee, & Mahmud, 2016; Wang et al., 2020; Weng, Lu, Wang, Megahed, & Martinez, 2018). Last, the predictive performance of RF and LSTM neural network is discussed. RF owns the lowest MSE and MAE among these models. Also, its standard deviation is the smallest. In addition, RF's H_R is pretty high among these models.

In conclusion, this section presents that RF outperforms the other models in stock return prediction process, and the predictive performance of LSTM neural network is the second followed by DMLP. Besides, the traditional time series model ARIMA performs better than CNN and SVR, and CNN's predictive error is the highest among these models. This result is similar to the conclusion in Ballings et al. (2015).

Table 8
The performance of different MVF models.

Model	ER	SD	IR	TOR	MD	TUR
DMLP+MVF	9.71%	0.7446	0.1304	7.88%	75.29%	54.21%
LSTM+MVF	46.66%	0.6685	0.6979	59.79%	74.75%	53.04%
CNN+MVF	39.39%	0.8541	0.4612	284.43%	52.99%	124.72%
SVR+MVF	97.34%	1.5395	0.6323	449.53%	69.79%	86.71%
RF+MVF	274.88%	3.3805	0.8131	1780.34%	59.61%	155.36%
ARIMA+MVF	20.70%	0.4202	0.4926	126.05%	64.54%	33.66%

ER, SD, IR, TOR, MD and TUR mean excess return, standard deviation, information ratio, total return, maximum drawdown and turnover rate respectively.

Table 9
The performance of different OF models.

Model	ER	SD	IR	TOR	MD	TUR
DMLP+OF	14.76%	0.9467	0.1560	−8.42%	77.63%	57.10%
LSTM+OF	120.63%	1.6467	0.7325	56.14%	87.51%	51.91%
CNN+OF	46.68%	0.8330	0.5604	314.43%	51.02%	126.19%
SVR+OF	133.18%	2.0262	0.6573	612.29%	74.36%	79.50%
RF+OF	121.53%	1.3980	0.8693	679.36%	70.42%	149.72%
ARIMA+OF	7.90%	0.3912	0.2020	51.96%	65.78%	40.39%

ER, SD, IR, TOR, MD and TUR mean excess return, standard deviation, information ratio, total return, maximum drawdown and turnover rate respectively.

This is mainly due to the input features used in this paper. Note that LSTM neural network may perform better than RF if more input features are contained.

5.2. Model performance without transaction fee

After comparing the performance of different models in stock return prediction, trading simulation is conducted to investigate the investing abilities of different portfolios. This paper simulates buying and selling behaviors as a typical investor. Specifically, an investor decides to buy or sell certain proportion of each stock from the market before each trading day to achieve the calculated proportion of each stock in the portfolio. For simplicity, the dividends and taxes are neglected, also leveraging and short selling are overlooked when investing. Also, the trading cost is not considered in this section. The trading simulation is implemented for all over the testing period, including 970 samples.

This paper applies six metrics, i.e., excess return, standard deviation, information ratio, total return, maximum drawdown and turnover rate to comprehensively evaluate the abilities of different portfolio models, where excess return means monthly average excess return and standard deviation measures the volatility of excess return each month. Total return represents the whole profit during the test period. The definitions of information ratio, maximum drawdown and turnover rate are presented as follows.

$$Information\ ratio = \frac{excess\ return}{standard\ deviation} \tag{40}$$

$$Maximum\ drawdown = \max_{l<s} \frac{Netv_l - Netv_s}{Netv_l} \tag{41}$$

$$Turnover\ rate = \sum_{i=1}^{n} |x_{i,t} - x_{i,t-1}| \tag{42}$$

where $Netv_l, x_{i,t}$ and n represent net value at time l, the weight of stock x in portfolio at time t and the number of stocks in portfolio. Note that, excess return, information ratio and total return are set as core metrics to compare different models since these metrics thoroughly represent their profitabilities.

5.2.1. MVF models

This section presents the experimental results of different MVF models. And, this paper uses RF +MVF to represent MVF model with RF forecasts, the others are similar.

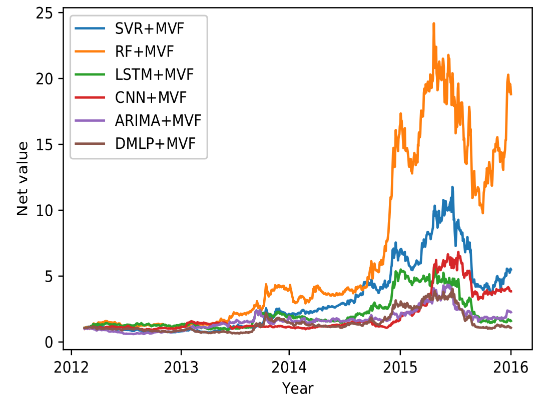

Fig. 1. Net value of different MVF models.

First, this paper compares three MVF models with deep learning models' forecasts. From Table 8, LSTM+MVF owns the highest excess return and information ratio, and CNN+MVF has the highest total return. Thus, CNN+MVF and LSTM+MVF perform better than DMLP+MVF. In the following, the performance of CNN+MVF and LSTM+MVF is further compared. Fig. 1 presents that there is no transparent difference between these models. In addition, Mann–Whitney test is conducted to compare their excess returns, test's $p-$ value equals to 0.216, which means there is no significant difference between these models statistically. Thus, CNN+MVF and LSTM+MVF have their own advantages and we cannot simply distinguish them.

Second, two MVF models with machine learning models' forecasts are compared. Table 8 shows that RF+MVF model owns higher excess return, information ratio and total return than SVR+MVF. Also, Fig. 1 depicts that the net value of RF+MVF is higher than SVR+MVF. Thus, RF+MVF outperforms SVR+MVF.

Third, CNN+MVF, LSTM+MVF, RF+MVF and ARIMA+MVF are further compared. From Table 8, RF+MVF has the highest excess return, information ratio and total return. Also, Fig. 1 presents that RF+MVF's net value is the largest among these models. Thus, RF+MVF is the best choice among the MVF models.

In addition, the monthly excess returns of different MVF models are presented in Figs. 2–5, which presents the performance of different models each month in detail. These figures display that the monthly excess returns of RF+MVF and LSTM+MVF in 2012 are the highest among these models and there is little difference between them. But from 2013 to 2015, the RF+MVF's excess return is almost the largest among these models.

Based on the above analysis, this paper concludes that RF+MVF model outperforms the other MVF models. Therefore, RF is more suitable to build MVF portfolio model.

5.2.2. OF models

This section discusses the performance of different OF models. Also, this paper uses RF+OF to denote OF model with RF forecasts, the others are similar.

First, this paper compares the performance of different OF models with deep learning models' forecasts. Table 9 shows that LSTM+OF owns the highest excess return and information ratio, and CNN+OF has the highest total return. Thus, LSTM+OF and CNN+OF perform better than DMLP+OF. Then, the performance of LSTM+OF and CNN+OF is further compared. Fig. 6 depicts that LSTM+OF's net value is larger than CNN+OF until the second half of 2015. In addition, Mann–Whitney test is conducted to compare their excess returns, test's $p-$ value equals to 0.001, which means there is significant difference between them in statistical sense. Thus, LSTM+OF outperforms CNN+OF.

Second, two OF models with machine learning models' forecasts are discussed. Table 9 shows that the difference of these models' excess

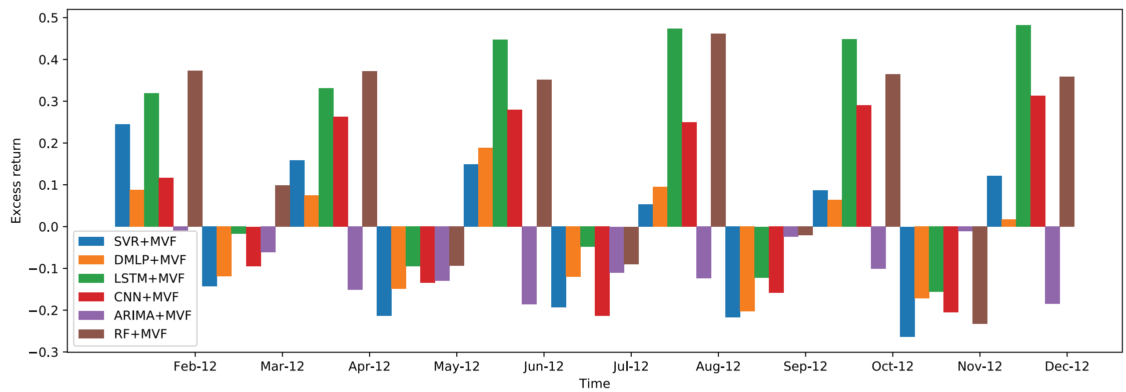

Fig. 2. Excess returns of different MVF models in 2012.

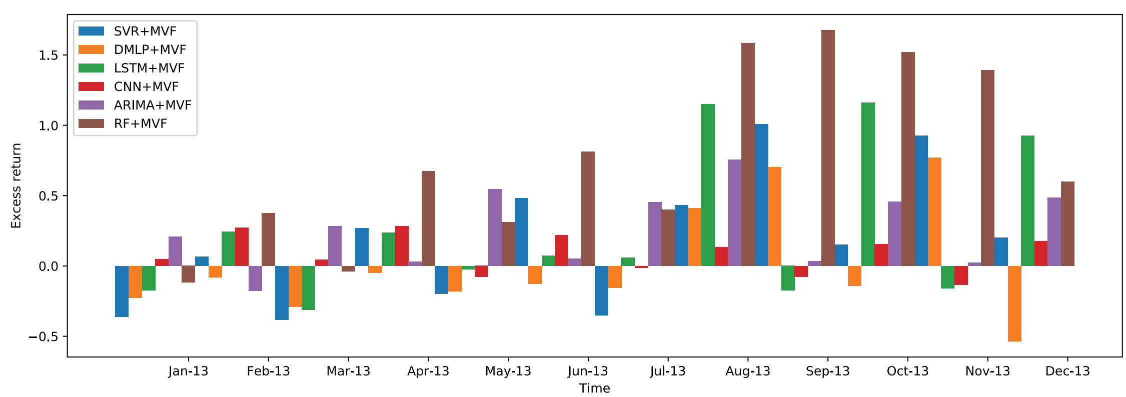

Fig. 3. Excess returns of different MVF models in 2013.

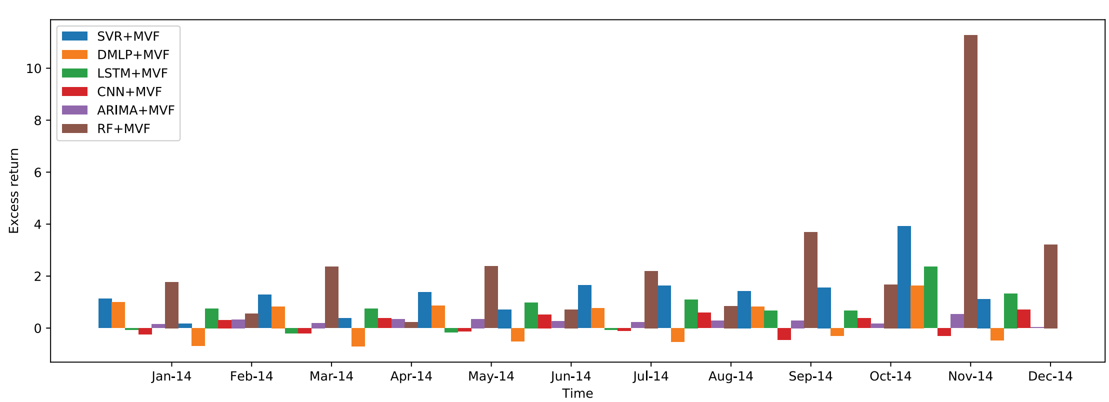

Fig. 4. Excess returns of different MVF models in 2014.

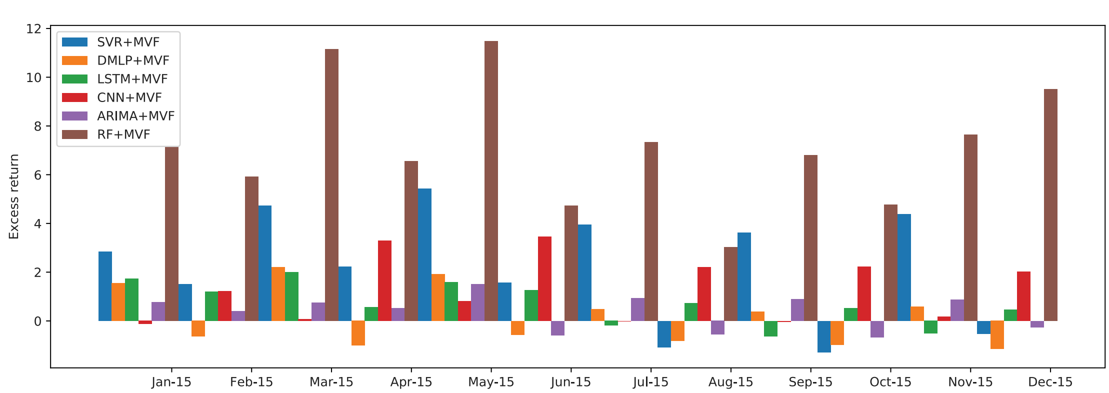

Fig. 5. Excess returns of different MVF models in 2015.

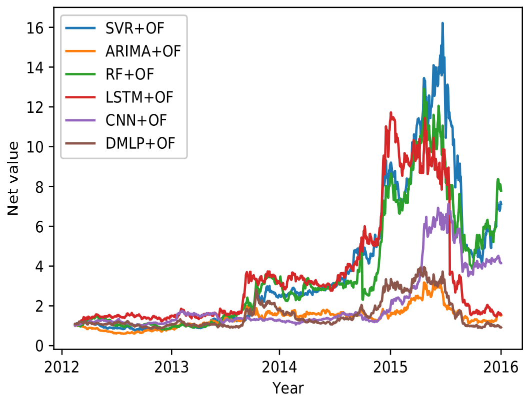

Fig. 6. Net value of different OF models.

returns is small and RF+OF's information ratio and total return are much higher than SVR+OF. Thus, RF+OF is a better choice for trading investment.

Third, the performance of RF+OF, LSTM+OF and ARIMA+OF is further compared. Table 9 presents that RF+OF owns the highest excess return, information ratio and total return among these models. Also, Fig. 6 shows that the net value of RF+OF is pretty large among these models. Therefore, RF+OF is the best choice among these models.

In addition, the monthly excess returns of different OF models are presented in Figs. 7–10. These figures show that LSTM+OF performs the best from 2012 to 2013, but since then it is gradually surpassed by SVR+OF. And, the performance of RF+OF is relatively stable and satisfying during the whole test period.

Last, since RF+MVF and RF+OF are the best models among all the considered MVF and OF models, it is necessary to compare these two models. Tables 8–9 present that RF+MVF's excess return and total return are much higher than RF+OF, and its information ratio is slightly lower than RF+OF. Thus, RF+MVF is superior to RF+OF.

Based on the analysis, this paper deduces that RF+OF performs the best among these OF models. And RF+MVF is superior to RF+OF. However, their turnover rates are both the highest among these models. Since high turnover causes high transaction fee, we deduct their transaction fees and further compare their profitabilities in the following section. In addition, the performance of different MVF generally corresponds to the performance of different OF models. This phenomenon is mainly attribute to the same way of combining return prediction in advancing MV and omega models. Moreover, RF+MVF and RF+OF perform the best among these models due to the optimal predictive ability of RF. However, the other portfolios' performance do not exactly match their predictive models. Possible reasons are the utilized predictive information in Eqs. (4)–(5).

5.3. Model performance with transaction fee

Based on the experiments of above section, we can obtain that RF+MVF and RF+OF have the highest turnover rates among MVF and OF models. As high turnover will increase transaction fee in real stock trading investment, it is necessary to explore the real performance of these models by deducting their transaction fees. Thus, this section discusses the performance of different models after deducting the transaction fee cased by turnover in order to further compare their profitabilities in real stock market. This paper uses the transaction fee caused by turnover of 0.05% per unit to approximate the total transaction fee of real trading investment for simplicity. Also, excess return, information ratio and total return are set as key metrics since these metrics comprehensively show their profitabilities.

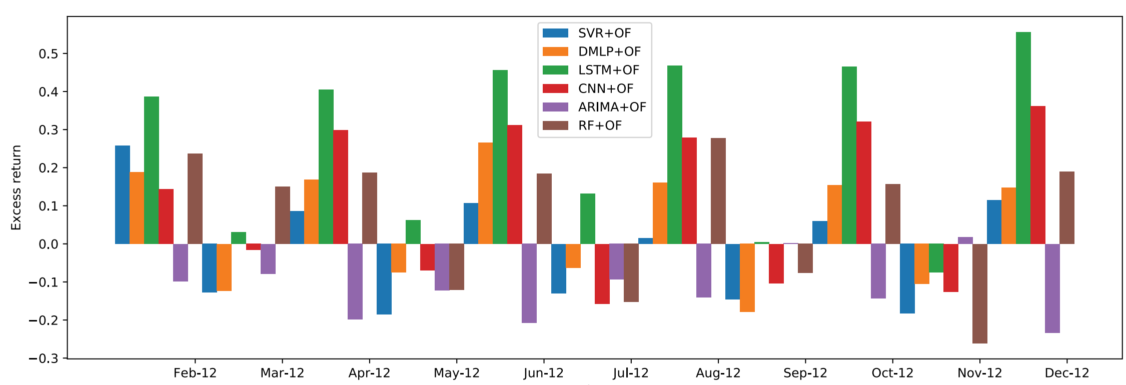

Fig. 7. Excess returns of different OF models in 2012.

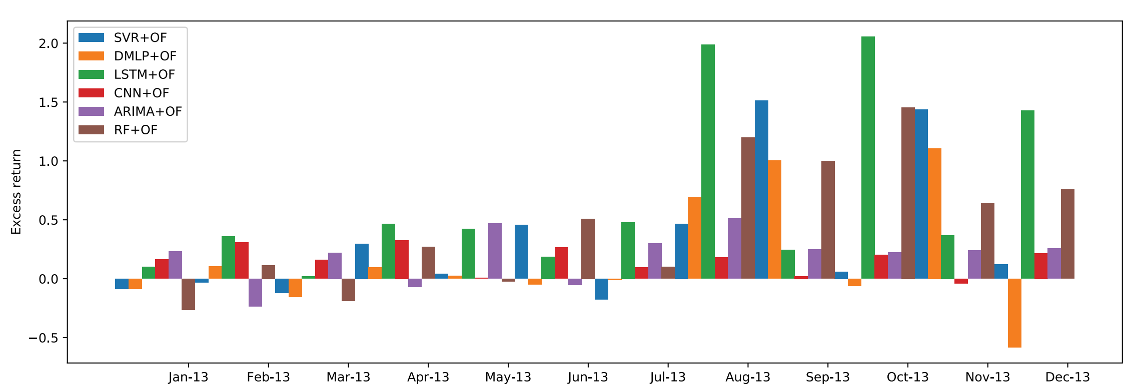

Fig. 8. Excess returns of different OF models in 2013.

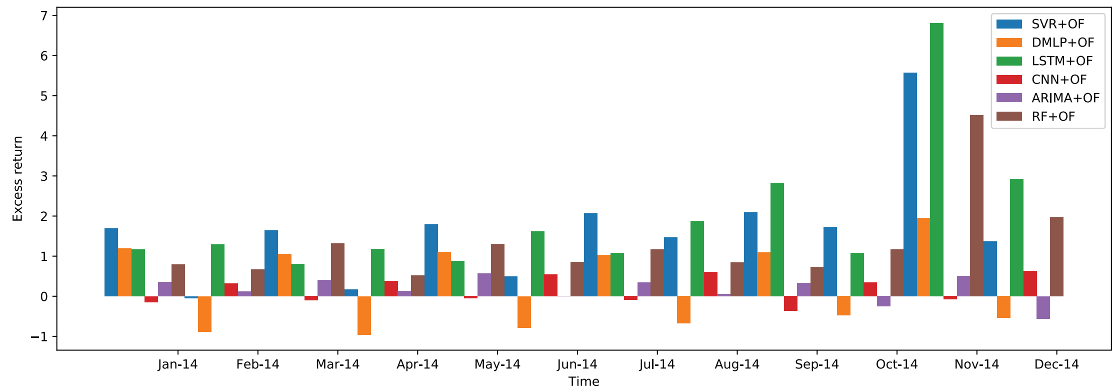

Fig. 9. Excess returns of different OF models in 2014.

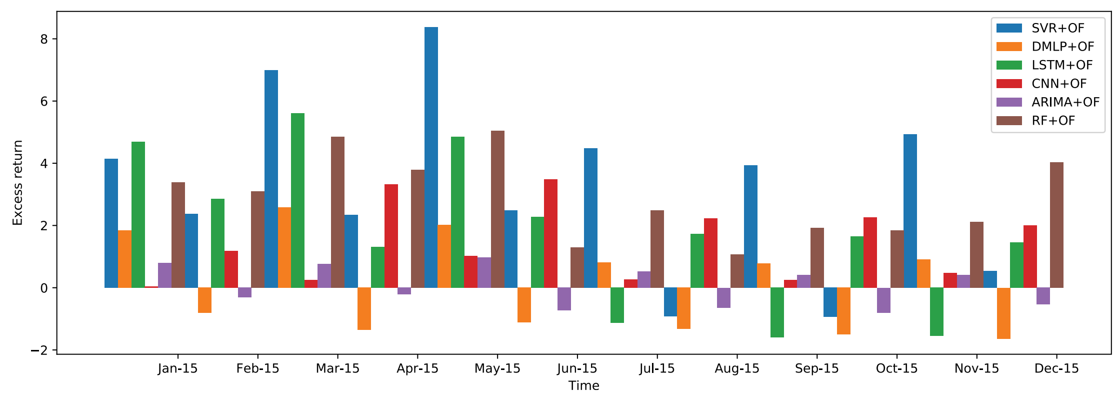

Fig. 10. Excess returns of different OF models in 2015.

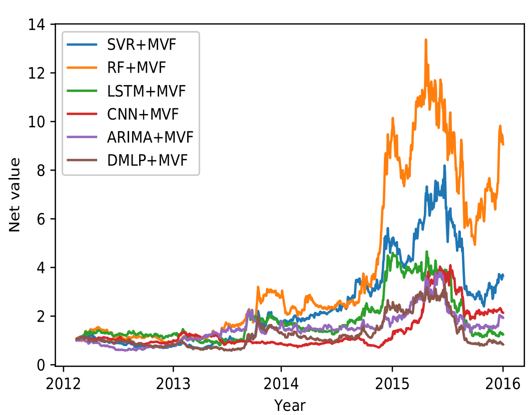

Fig. 11. Net value of different MVF models with transaction fee.

Table 10
The performance of different MVF models with transaction fee.

Model	ER	SD	IR	TR	MD
DMLP+MVF	−0.10%	0.6312	−0.0015	−16.42%	76.15%
LSTM+MVF	31.60%	0.5617	0.5625	24.43%	75.73%
CNN+MVF	7.17%	0.5540	0.1295	114.07%	54.50%
SVR+MVF	58.34%	1.0327	0.5650	265.80%	70.71%
RF+MVF	134.76%	1.6835	0.8005	806.52%	63.10%
ARIMA+MVF	13.01%	0.3491	0.3726	93.01%	64.95%

ER, SD, IR, TOR and MD mean excess return, standard deviation, information ratio, total return and maximum drawdown respectively.

5.3.1. *MVF models with transaction fee*

This section discusses the performance of different MVF models with transaction fee.

First, three MVF models with deep learning models' forecasts are compared. From Table 10, LSTM+MVF owns the highest excess return and information ratio, and CNN+MVF has the largest total return. Thus, LSTM+MVF and CNN+MVF perform better than DMLP+MVF. In the following, LSTM+MVF and CNN+MVF are further compared. Mann–Whitney test is conducted to compare their excess returns, test's $p-$ value equals to 0.012, which means there is significant difference between them. Thus, LSTM+MVF performs better than CNN+MVF.

Second, two MVF models with machine learning models' forecasts are compared. Table 10 shows that RF+MVF's excess return, information ratio and total return are higher than SVR+MVF. Also, Fig. 11 depicts that the net value of RF+MVF is significantly higher than SVR+MVF model. Thus, RF+MVF is a better choice than SVR+MVF.

Third, this paper compares the differences between LSTM+MVF, RF+MVF and ARIMA+MVF. From Table 10, RF+MVF owns the highest excess return, information ratio and total return. Also, Fig. 11 displays that RF+MVF's net value is the largest among these models. Therefore, RF+MVF performs the best among these models.

Figs. 12–15 present the monthly excess returns of different MVF models after deducting their transaction fees caused by turnover. These figures show that the superiority of RF+MVF is further improved compared with other models. In conclusion, RF+MVF outperforms the other models, and high turnover erodes nearly half of its total return. Thus, this paper recommends to build MVF model with RF forecasts and it is necessary to consider transaction fee when testing the performance of different models.

5.3.2. *OF models with transaction fee*

This section discusses different OF models after deducting their transaction fees.

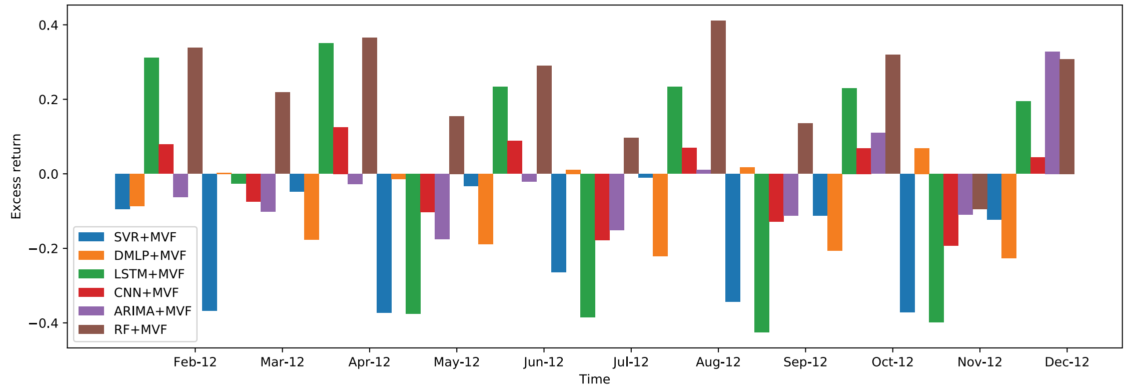

Fig. 12. Excess return of different MVF models with transaction fee in 2012.

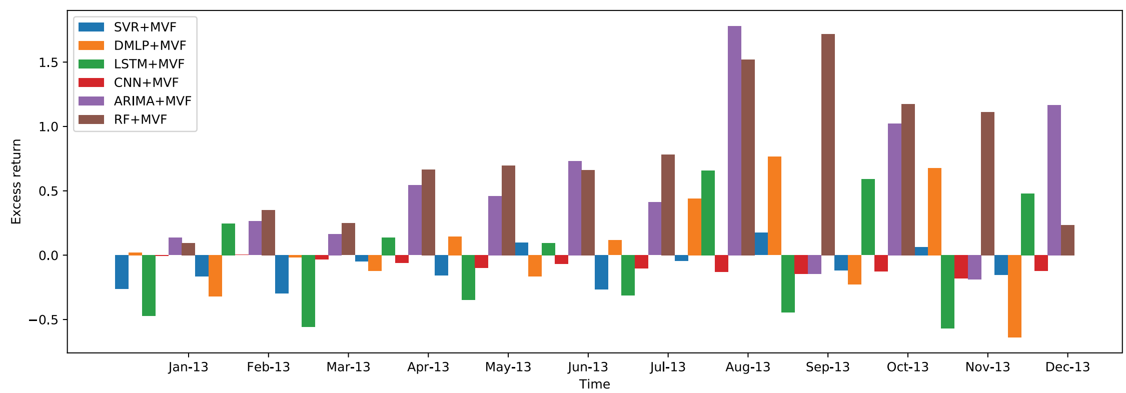

Fig. 13. Excess return of different MVF models with transaction fee in 2013.

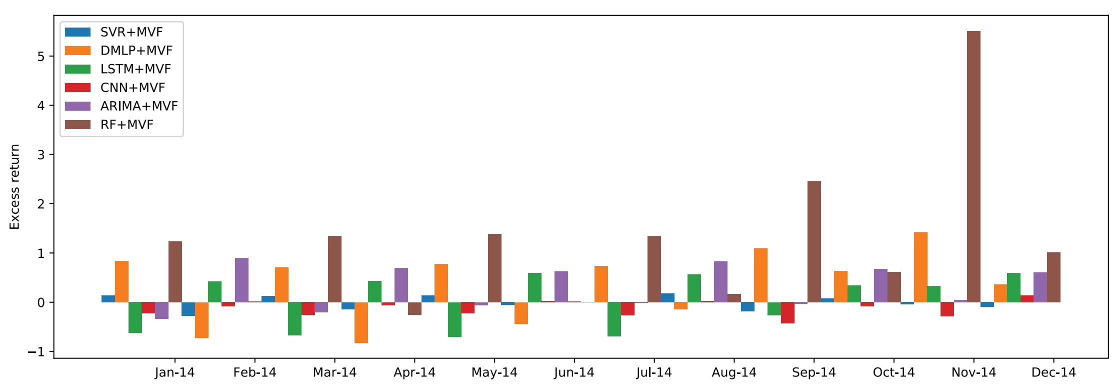

Fig. 14. Excess return of different MVF models with transaction fee in 2014.

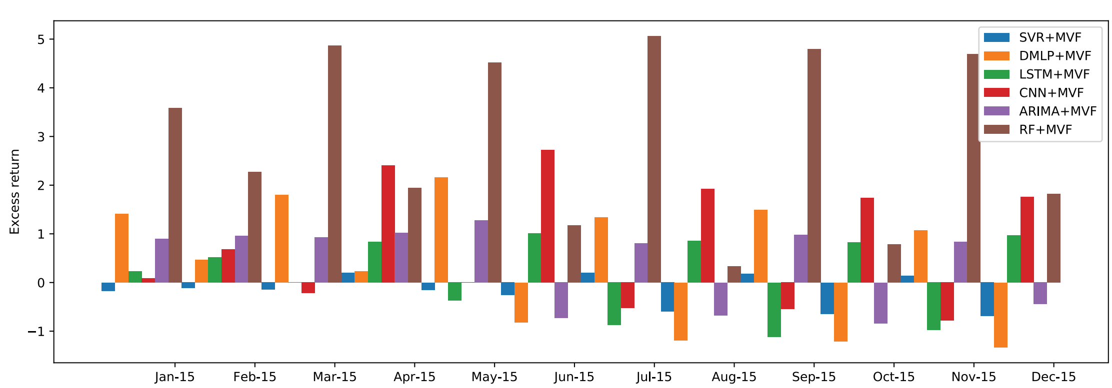

Fig. 15. Excess return of different MVF models with transaction fee in 2015.

Table 11
The performance of different OF models with transaction fee.

Model	ER	SD	IR	TR	MD
DMLP+OF	4.07%	0.8060	0.0505	−30.00%	78.69%
LSTM+OF	95.31%	1.3508	0.7056	22.21%	88.37%
CNN+OF	12.07%	0.5167	0.2335	129.23%	52.67%
SVR+OF	87.56%	1.4373	0.6092	390.46%	74.93%
RF+OF	50.87%	0.6532	0.7788	285.40%	72.82%
ARIMA+OF	0.54%	0.3433	0.0158	25.73%	66.34%

ER, SD, IR, TOR and MD mean excess return, standard deviation, information ratio, total return and maximum drawdown respectively.

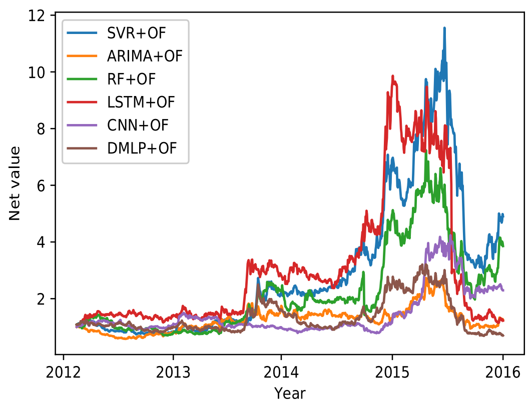

Fig. 16. Net value of different OF models with transaction fee.

First, three OF models with deep learning models' forecasts are compared. From Table 11, LSTM+OF owns the highest excess return and information ratio, and CNN+OF has the largest total return. Thus, LSTM+OF and CNN+OF outperform DMLP+OF. Then, LSTM+OF and CNN+OF are further compared. Fig. 16 depicts that the net value of LSTM+OF is higher than CNN+OF until the second half of 2015. In addition, Mann–Whitney test is conducted to measure their excess returns, test's result shows that $p-$ value equals to 0.000, which means there is significant difference between them. Thus, LSTM+OF performs better than CNN+OF.

Second, two OF models with machine learning models' forecasts are discussed. Table 11 gives that SVR+OF has higher excess return and total return, and RF+OF has larger information ratio. Thus, it is difficult to differentiate them with three core metrics. Further, according to Fig. 16, SVR+OF's net value is larger than RF+OF model in most cases. Therefore, SVR+OF performs better than RF+OF. Note that, RF+OF's turnover rate is nearly twice that of SVR+OF, which causes its poor performance after deducting transaction fee.

Third, LSTM+OF, SVR+OF and ARIMA+OF are compared. Table 11 presents that LSTM+OF owns the highest excess return, information ratio, and SVR+OF has the largest total return. Thus, LSTM+OF and SVR+OF outperform ARIMA+OF. Then, LSTM+OF and SVR+OF are further compared. Fig. 16 shows that the net value of LSTM+OF is the largest in 2012–2015 and then it is surpassed by SVR+OF in 2015. Also, since SVR+OF not only owns similar excess return and information ratio to LSTM+OF but also has much larger total return than LSTM+OF. Therefore, SVR+OF is a better choice than LSTM+OF.

Moreover, the monthly excess returns of different OF models are presented in Figs. 17–20, which presents that the relative performance of different models do not change after deducting their transaction fees.

Last, this paper compares RF+MVF with SVR+OF. Tables 10–11 show that RF+MVF owns higher excess return, information ratio and total return than SVR+OF. Therefore, RF+MVF outperforms SVR+OF. In conclusion, SVR+OF performs the best among OF models after deducting the transaction fee. And turnover erodes nearly half of RF+OF's total profit, which greatly influences its profitability. Thus, this paper suggests to build omega model with SVR prediction for trading investment.

6. Discussion and conclusion

6.1. Discussion of key findings

This study aims to extend the existing literature on portfolio construction with return prediction. Two machine learning models and three deep learning models are applied to advance the MV and omega models, which combines the advantages of machine learning and deep learning models in portfolio formation. The test period ranges from January 5, 2012 to December 31, 2015, containing 970 days, and the experiment focuses on Chinese stock market, i.e., the China Securities 100 Index component stocks.

First, this paper compares the predictive abilities of RF, SVR, DMLP, LSTM neural network and CNN in stock return prediction. Experimental results show that RF outperforms the other models. Second, this paper discusses the performance of different MVF and OF models without transaction fee, and applies six metrics to comprehensively measure their differences. Experiments' results present that RF+MVF and RF+OF perform the best among these models. Further, RF+MVF is superior to RF+OF. However, the main defect of RF+MVF and RF+OF is high turnover, which may erode considerable total profit. Thus, this paper further compares the performance of different models after deducting their transaction fees. Experimental results show that RF+MVF still outperforms the other MVF models, but RF+OF is surpassed by SVR+OF since it owns higher turnover rate. And, the RF+MVF model performs better than SVR+OF. In addition, turnover erodes nearly half of their total returns especially for RF+MVF and RF+OF. Therefore, this paper recommends to build MVF model with RF return forecasts for daily trading investment.

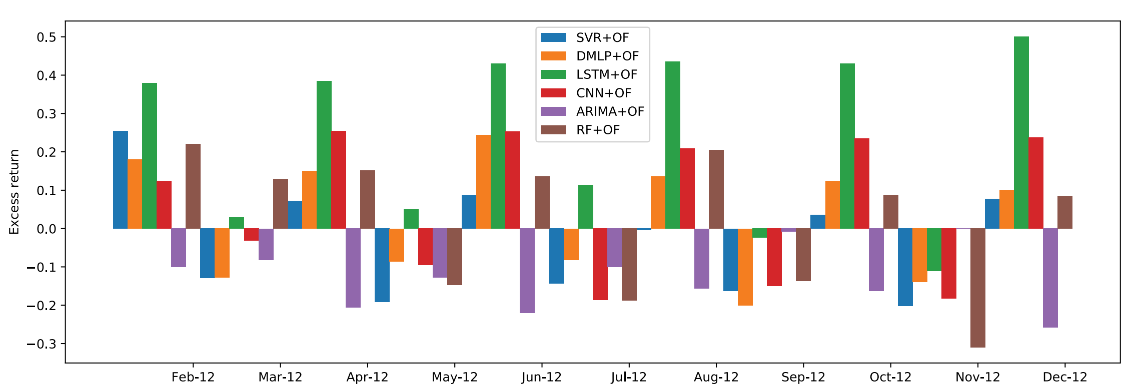

Fig. 17. Excess return of different OF models with transaction fee in 2012.

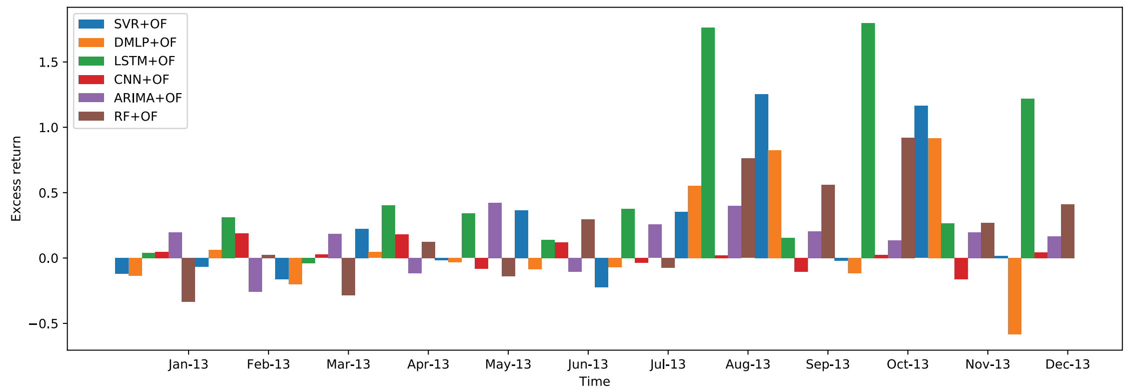

Fig. 18. Excess return of different OF models with transaction fee in 2013.

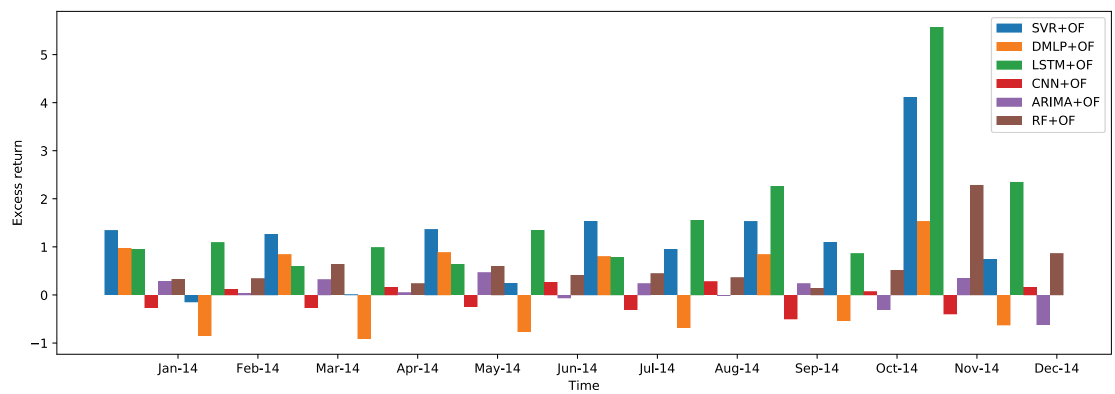

Fig. 19. Excess return of different OF models with transaction fee in 2014.

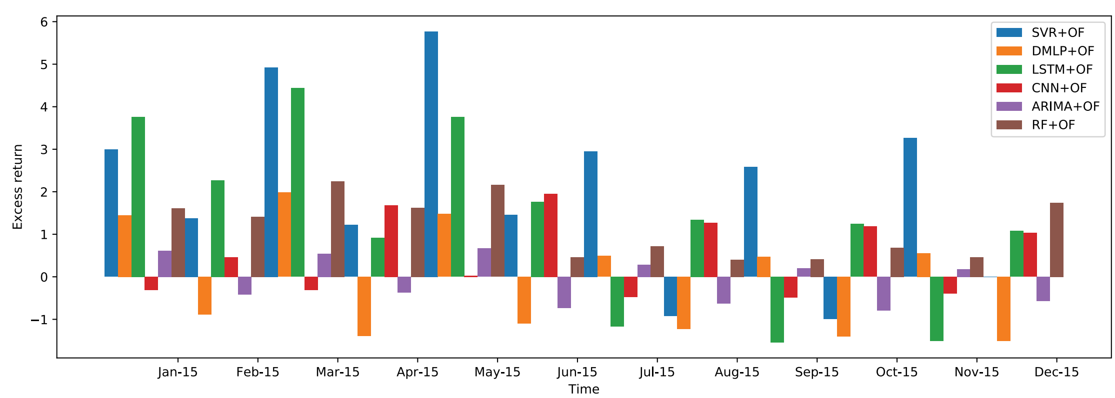

Fig. 20. Excess return of different OF models with transaction fee in 2015.

6.2. Theoretical implications

This study enriches the theoretical researches on portfolio optimization with return prediction. First of all, this paper uses five models in stock return prediction process, which guarantees the high-quality stocks are selected before building portfolio optimization models. To be specific, RF, SVR, DMLP, LSTM neural network and CNN are adopted for future daily return prediction, and ARIMA model is used as benchmark to show their superiorities. Second, two valuable portfolio optimization models are advanced with above models' return predictive results, which fills the research gap in existing literature. Actually, MV and omega models are extended with deep learning and machine learning models' predictive results for the first time, and portfolio optimization models with ARIMA return prediction are used as comparisons to present their advantages.

6.3. Limitations and future work

This study also has limitations because we only apply simple historical returns as input features in order to compare with the benchmark model. Since many studies have shown the value of technical indicators, news, exchange rate and economic indicators. Thus, further studies can try to apply more efficient input features to train predictive models and improve the performance of MVF and OF models for daily trading investment. Also, high turnover is a big challenge for these models to overcome especially for RF+MVF, which is profitable in practical investment transaction.

CRediT authorship contribution statement

Yilin Ma: Conceptualization, Data curation, Formal analysis, Investigation, Methodology, Software, Validation, Visualization, Writing

- original draft, Writing - review & editing. **Ruizhu Han:** Supervision, Funding acquisition, Project administration, Resources, Writing - review & editing. **Weizhong Wang:** 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.

Acknowledgments

Yilin Ma conceptualized the research, collected and analyzed the data, and conducted the experiment. Yilin Ma, Ruizhu Han and Weizhong Wang wrote, reviewed and edited the paper.

Funding

This research was supported by the National Natural Science Foundation of China (No. 71390335).

References

Adebiyi, A. A., Adewumi, A. O., & Ayo, C. K. (2014). Comparison of ARIMA and artificial Neural Networks models for Stock Price Prediction. *Journal of Applied Mathematics*, http://dx.doi.org/10.1155/2014/614342.

Alexander, G. J., & Baptista, A. M. (2002). Economic implications of using a mean-VaR model for portfolio selection: A comparison with mean–variance analysis. *Journal of Economic Dynamics and Control*, *26*, 1159–1193.

Alizadeh, M., Rada, R., Jolai, F., & Fotoohi, E. (2011). An adaptive neuro-fuzzy system for stock portfolio analysis. *International Journal of Intelligent Systems*, *26*, 99–114.

Ballings, M., Poel, D. V. D., Hespeels, N., & Gryp, R. (2015). Evaluating multiple classifiers for stock price direction prediction. *Expert Systems with Applications*, *42*, 7046–7056.

Booth, A., Gerding, E., & Mcgroarty, F. (2014). Automated trading with performance weighted random forests and seasonality. *Expert Systems with Application*, *41*, 3651–3661.

Breiman, L. (2001). Random forests. *Machine Learning*, *45*, 5–32.

Chen, B. L., Zhong, J. D., & Chen, Y. Y. (2020). A hybrid approach for portfolio selection with higher-order moments: Empirical evidence from Shanghai Stock Exchange. *Expert Systems with Applications*, *145*, Article 113104.

Chong, E., Han, C., & Park, F. C. (2017). Deep learning networks for stock market analysis and prediction: Methodology, data representations, and case studies. *Expert Systems with Applications*, *83*, 187–205.

Deboeck, G. J. (1994). *Trading on the edge: Neural, genetic, and fuzzy systems for chaotic financial markets*. New York: Wiley.

Deng, G. F., Lin, W. T., & Lo, C. C. (2012). Markowitz-based portfolio selection with cardinality constraints using improved particle swarm optimization. *Expert Systems with Applications*, *39*, 4558–4566.

Deng, S. J., & Min, X. Y. (2013). Applied optimization in Global Efficient Portfolio Construction using Earning Forecasts. *The Journal of Investing*, *22*, 104–114.

Emir, Ş. (2013). Predicting the Istanbul Stock Exchange index return using Technical Indicators: A comparative study. *International Journal of Finance & Banking Studies*, *2*(3), 111–117.

Fabozzi, F. J., Gupta, F., & Markowitz, H. M. (2002). The legacy of modern portfolio theory. *The Journal of Investing*, *11*, 7–22.

Fischer, T., & Krauss, C. (2018). Deep learning with long short-term memory networks for financial market predictions. *European Journal of Operational Research*, *270*, 654–669.

Freitas, F. D., Souza, A. F. D., & Almeida, A. R. D. (2009). Prediction-based portfolio optimization model using neural networks. *Neurocomputing*, *72*, 2155–2170.

Gandhmal, D. P., & Kumar, K. (2019). Systematic analysis and review of stock market prediction techniques. *Computer Science Review*, *34*, Article 100190.

Gilli, M., Schumann, E., di Tollo, G., & Cabej, G. (2011). Constructing 130/30-portfolios with the Omega ratio. *Journal of Asset Management*, *12*, 94–108.

Graves, A., & Schmidhuber, J. (2005). Framewise phoneme classification with bidirectional lstm and other neural network architectures. *Neural Networks*, *18*, 602–610.

Hansen, J. V., & Nelson, R. D. (2002). Data mining of time series using stacked generalizers. *Neurocomputing*, *43*, 173–184.

Hao, C. Y., Wang, J. Q., & Xu, W. (2013). Prediction-based portfolio selection model using support vector machines. In *Proceedings of sixth international conference on business intelligence and financial engineering* (pp. 567–571).

Hiransha, M., Gopalakrishnan, E. A., Menon, V. K., & Soman, K. P. (2018). NSE Stock Market Prediction using Deep-learning Models. *Procedia Computer Science*, *132*, 1351–1362.

Ho, T. K. (1995). Random decision forests. In *Proceedings of the third international conference on document analysis and recognition* (pp. 278–282).

Hoseinzade, E., & Haratizadeh, S. (2019). CNNpred: CNN-based stock market prediction using a diverse set of variables. *Expert Systems with Applications*, *129*, 273–285.

Huang, C. F. (2012). A hybrid stock selection model using genetic algorithms and support vector regression. *Applied Soft Computing*, *12*, 807–818.

Ji, S. W., Xu, W., Yang, M., & Yu, K. (2012). 3D convolutional neural networks for human action recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *35*, 221–231.

Jorion, P. (1997). *Value at risk: The new benchmark for controlling market risk*. Chicago: McGraw-Hill.

Kane, S. J., Bartholomew-Biggs, M. C., Cross, M., & Dewar, M. (2009). Optimizing Omega. *Journal of Global Optimization*, *45*, 153–167.

Kapsos, M., Christofides, N., & Rustem, B. (2014). Worst-case robust Omega ratio. *European Journal of Operational Research*, *234*, 499–507.

Keating, C., & Shadwick, W. F. (2002). A universal performance measure. *Journal of Performance Measurement*, *6*, 59–84.

Kolm, P. N., Tütüncü, R., & Fabozzi, F. J. (2014). 60 Years of portfolio optimisation: Practical challenges and current trends. *European Journal of Operational Research*, *234*, 356–371.

Konno, H., & Yamazaki, H. (1991). Mean-absolute Deviation Portfolio Optimization model and its applications to Tokyo Stock Market. *Management Science*, *37*(5), 519–531.

Krauss, C., Do, X. A., & Huck, N. (2017). Deep neural networks, gradient-boosted trees, random forests: Statistical arbitrage on the S & P 500. *European Journal of Operational Research*, *259*, 689–702.

LeCun, Y., & Bengio, Y. (1995). Convolutional networks for images, speech, and time series. In *The handbook of brain theory and neural networks, Vol. 3361*. 1995.

Lee, S. I., & Yoo, S. J. (2018). Threshold-based Portfolio: The Role of the Threshold and its applications. *The Journal of Supercomputing*, http://dx.doi.org/10.1007/s11227-018-2577-1.

Lin, C. M., Huang, J. J., Gen, M., & Tzeng, G. H. (2006). Recurrent neural network for dynamic portfolio selection. *Applied Mathematics and Computation*, *175*, 1139–1146.

Long, W., Lu, Z., & Cui, L. (2018). Deep learning-based feature engineering for stock price movement prediction. *Knowledge-Based Systems*, *164*, 163–173.

Long, J., Shelhamer, E., & Darrell, T. (2015). Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 3431–3440).

Lu, C. J., Lee, T. S., & Chiu, C. C. (2009). Financial time series forecasting using independent component analysis and support vector regression. *Decision Support Systems*, *47*, 115–125.

Markowitz, H. M. (1959). *Portfolio selection: Efficient diversification of investments*. New York: John Wiley Sons Inc.

Matías, J. M., & Reboredo, J. C. (2012). Forecasting performance of nonlinear models for Intraday Stock Returns. *Journal of Forecasting*, *31*, 172–188.

Moews, B., Herrmann, J. M., & Ibikunle, G. (2019). Lagged correlation-based deep learning for directional trend change prediction in financial time series. *Expert Systems with Applications*, *120*, 197–206.

Oliveira, N., Cortez, P., & Areal, N. (2017). The impact of microblogging data for stock market prediction: Using Twitter to predict returns, volatility, trading volume and survey sentiment indices. *Expert Systems with Applications*, *73*, 125–144.

Orimoloye, L. O., Sung, M. C., Ma, T. J., & Johnson, J. E. V. (2020). Comparing the effectiveness of deep feedforward neural networks and shallow architectures for predicting stock price indices. *Expert Systems with Application*, *139*, Article 112828.

Paiva, F. D., Cardoso, R. T. N., Hanaoka, G. P., & Duarte, W. M. (2019). Decision-making for financial trading: A fusion approach of machine learning and portfolio selection. *Expert Systems with Applications*, *115*, 635–655.

Pang, X., Zhou, Y., Wang, P., Lin, W. W., & Chang, V. (2018). An innovative neural network approach for stock market prediction. *The Journal of Supercomputing*, http://dx.doi.org/10.1007/s11227-017-2228-y.

Patel, J., Shah, S., & Thakkar, P. (2015). Predicting stock and stock price index movement using Trend Deterministic Data Preparation and machine learning techniques. *Expert Systems with Applications*, *42*, 259–268.

Principe, J. C., Euliano, N. R., & Lefebvre, W. C. (1999). *Neural and adaptive systems: Fundamentals through simulations*. New York: Wiley.

Qin, Q. (2014). Linear and nonlinear Trading models with Gradient Boosted random Forests and Application to Singapore Stock Market. *Journal of Intelligent Learning Systems and Applications*, *5*, 1–10.

Rasel, R. I., Sultana, N., & Meesad, P. (2015). An efficient modelling approach for forecasting financial time series data using support vector regression and windowing operators. *International Journal of Computational Intelligence Studies*, *4*(2), 134–150.

Rockafellar, R. T., & Uryasev, S. (2000). Optimization of conditional value-at-risk. *Journal of Risk*, *2*, 21–42.

Sadaei, H. J., Enayatifar, R., Lee, M. H., & Mahmud, M. (2016). A hybrid model based on differential fuzzy logic relationships and imperialist competitive algorithm for stock market forecasting. *Applied Soft Computing*, *40*, 132–149.

Sezer, O. B., Gudelek, U., & Ozbayoglu, M. (2020). Financial time series forecasting with deep learning : A systematic literature review: 2005-2019. *Applied Soft Computing*, *90*, Article 106181.

Sezer, O. B., & Ozbayoglu, A. M. (2018). Algorithmic Financial Trading with deep Convolutional Neural Networks: Time Series to image Conversion approach. *Applied Soft Computing*, *70*, 525–538.

Singh, R., & Srivastava, S. (2017). Stock prediction using deep learning. *Multimedia Tools and Applications*, *76*, 18569–18584.

Ta, V. D., Liu, C. M., & Addis, D. (2018). Prediction and portfolio optimization in quantitative trading using machine learning techniques. In *Proceedings of the ninth international symposium on information and communication technology* (98–105).

Ta, V. D., Liu, C. M., & Tadesse, D. A. (2020). Portfolio optimization-based stock prediction using long-short term memory network in quantitative trading. *Applied Sciences*, *10*, 437.

Ustun, O., & Kasimbeyli, R. (2012). Combined forecasts in portfolio optimization: A generalized approach. *Computers & Operations Research*, *39*, 805–819, 2012.

Wang, W., Li, W., Zhang, N., & Liu, K. C. (2020). Portfolio formation with preselection using deep learning from long-term financial data. *Expert Systems with Applications*, *143*, Article 113042.

Wang, J. Z., Wang, J. J., Zhang, Z. G., & Guo, S. P. (2011). Forecasting stock indices with back propagation neural network. *Expert Systems with Applications*, *38*, 14346–14355.

Weng, B., Lu, L., Wang, X., Megahed, F. M., & Martinez, W. (2018). Predicting short-term stock prices using ensemble methods and online data sources. *Expert Systems with Applications*, *112*, 258–273.

Yu, J. R., Chiou, W. J. P., Lee, W. Y., & Lin, S. J. (2020). Portfolio models with return forecasting and transaction costs. *International Review of Economics & Finance*, *66*, 118–130, 2020.

Zhang, Y., Li, X., & Guo, S. (2018). Portfolio selection problems with Markowitz's mean–variance framework: A review of literature. *Fuzzy Optimization and Decision Making*, *17*, 125–158.

Zhang, Y., & Wu, L. (2009). Stock market prediction of S & P 500 via combination of improved BCO approach and BP neural network. *Expert Systems with Applications*, *36*, 8849–8854.

Zhu, M. (2013). Return distribution predictability and its implications for portfolio selection. *International Review of Economics & Finance*, *27*, 209–223.