# Long Short-Term Memory-based Multi-Period Price Prediction for Portfolio Management

Zhengyong Jiang[1,2,4] and Frans Coenen[3]

[1] Department of Electrical and Electronic Engineering, University of Liverpool based in Xi'an Jiaotong-Liverpool University, Suzhou, China
[2] Department of Mathematical Sciences, Xi'an Jiaotong-Liverpool University, Suzhou, China
[3] Department of Computer Science, University of Liverpool, Liverpool, UK
[4] Authors to whom correspondence should be addressed
zhengyong.jiang@xjtlu.edu.cn

**Abstract.** In this paper, we present a novel, machine learning-based approach for price prediction to portfolio construction in the context of multi-period trading. We use a combination of recurrent neural network (RNN) and a long short-term memory (LSTM) network for predicting the future prices and to perform constrained optimization on the portfolio update.
Evaluation using a series of back-test on a number of datasets obtained from Shanghai Stock Exchange (SSE) and National Association of Securities Deal Automated Quotations (NASDAQ) sources show that our proposed approach outperforms the most common conventional portfolio management technique, namely Robust Median Reversion strategy, on a number of different metrics. In back-test experiment, our proposed method offers an average of 148% returns over 360 trading periods with 100 stocks, compared to 124% returns using conventional technique, over the same period and number of stocks.

**Keywords:** Portfolio Management, Recurrent Neural Network, Long Short-Term Memory

## 1 Introduction

In financel context, the term "portfolio" refers to any combination of financial assets such as stocks, bonds and cash which are held by individual investors or managed by financial professionals, hedge funds, banks and other financial institutions [30]. Portfolio management is the decision-making process of allocating a specified amount of fund to a set of different assets, with the aim of maximizing the return under the same level of risk [5, 9]. This can be seen as a constrained optimization problem. To this end, the mean-variance model and the principle of efficient frontier are two most common techniques used to construct portfolios [30].

Owing to the underlying mathematical model, the mean-variance model is only suitable for single-period trading, and extending that to update multiple periods results in computationally intractable solution. Moreover, nearly all of the tradings in the stock markets are multi-period [13]. As such, techniques based on mean-variance model are less useful in the context of multi-period trading.

In the recent past, machine learning-based approaches have become a popular choice for solving a number of financial, particularly portfolio management, problems [34], in addition to solving a number of problems from the signal processing domain [38, 40]. One of the common principles of machine learning is to supervise the machine to learn by examples — supervised machine learning. An underpinning aspect of learning here, in conventional methods, is identifying the features to learn. This process, often referred to as feature engineering, is central to most of the supervised learning. Among different machine learning-based approaches, deep learning has become a popular method for addressing a number of problems. Deep learning methods rely on neural networks and their variants to learn features themselves along with the task. As such, deep learning-based techniques have become an attractive choice for a number of tasks both for classification problems and for regression problems.

In the context of price prediction, the historical stock prices are treated as a time series and future prices are predicted by learning from the past. The novel technique we propose here in this paper combines deep learning method and constrained optimization for handling the portfolio construction problem in the context of multi-period trading. In particular, we use a variant of neural network called Recurrent Neural Network (RNN) along with another variant called Long Short Term Memory network (RNN-LSTM) to predict the future price of each asset and to update the concerning portfolio. The combination of networks we use in our method not only learns the features that are responsible for the fluctuations in price, but also remembers the past due to its long-term memory. The key contributions of this paper are two fold:

1. We propose a novel price prediction based on a combination of RNN-LSTM network that offers far superior results than the conventional Robust Median Reversion (RMR) method; and
2. We perform a thorough evaluation of our approach and validate its effectiveness by performing back-test on 100 real-world stocks. We compare our strategy against other strategies such as Passive Aggressive Median Reversion (PAMR) and Confidence Weighted Median Reversion Strategy (CWMR).

The rest of the paper is organized as follows. The background researches about portfolio construction is discussed in Section 2. In Section 3, we discuss our approach in solving portfolio management problem. The results of back-tests are then discussed in Section 4. Finally, we conclude the paper with directions for further research in Section 5.

## 2 Background

### 2.1 Problem Statement

For a given financial market, suppose that we are interested in the investment of $d$ assets for $n$ trading days altogether. At the beginning of $t^{\text{th}}$ trading day, our investment for the $d$ assets is denoted by the portfolio vector $\mathbf{b}_t = \left[b_t^1, \ldots, b_t^d\right]^T$ where $b_t^j \in [0,1]$ represents the proportion of wealth invested in the asset $j \in \{1,2,\ldots,d\}$ where $b_t^1 + b_t^2 + \ldots + b_t^d = 1$. Following the investment, let vector $\mathbf{p}_t = \left[p_t^1, p_t^2, \ldots, p_t^d\right] \in \mathbb{R}_+^d$ represent the close price of all $d$ assets at the end of $t^{th}$ trading days. The vector $\mathbf{x}_t = \left[x_t^1, \ldots, x_t^d\right]^T \in \mathbb{R}_+^d$ gives the ratio of current close price to the previous close price for each asset $j \in \{1,2,\ldots,d\}$ at time $t$, i.e., $x_t^j = p_t^j / p_{t-1}^j$. At the end of $t^{th}$ trading day, we achieve a period return $S_t = \mathbf{b}_t^T \mathbf{x}_t = \sum\limits_{j=1}^{d} b_t^j x_t^j$. The aim of portfolio management is to design a strategy for determining the portfolio vector $\mathbf{b}_t$ at the beginning of $t^{th}$ trading day so as to maximize the final cumulative portfolio wealth $S_n = S_0 \prod_{t=1}^{n} (\mathbf{b}_t^T \mathbf{x}_t)$ where $S_0$ is the initial wealth at the beginning of trading. The strategy shall be measured based on the final cumulative portfolio wealth and other metrics which are introduced later in this paper.

In this research, we shall assume that the market is in a perfect liquidity, with zero impact cost situation, and the price of each stock is independent from one another. These assumptions are not trivial. The first assumption ensures that we could invest our capital in each asset with any possible proportion. The second assumption ensures that we could obtain price information immediately at any time nodes without any cost. The third assumption enables us to predict future price of each stock independently of others.

### 2.2 Related work

There are two main mathematical principles in the portfolio management problem, namely the efficient frontier principle [30] and the Kelly investment principle [18]. The mean-variance model, which is based on the efficient frontier principle, trades in the market according to expected return of the stock and risk (i.e., variance of the price of the stock) [3], is suitable for single-period portfolio management. The Kelly investment principle, which targets to maximize the expected return, focuses on multiple-period sequential portfolio management.

Traditional multiple-period sequential portfolio management methods can be classified into four categories, namely, the *Follow-the-winner, Follow-the-loser, Pattern-Matching and Meta-Learning* [24]. The first two categories are based on existing financial models such as mean reversion model and exponential gradient model. The *Follow-the-winner* algorithm is inclined to invest stocks in a upward trend while the *Follow-the-loser* algorithm is inclined to invest stocks in a downward trend. They may also be assisted by some online learning techniques, e.g., statistical techniques to improve the performance [20] and to tune the parameter in them [23] [7]. The performance of these methods depends on the validity of models on different markets such as stock markets, futures markets or cryptocurrency markets. The *Pattern-Matching* algorithm selects part of historical data similar to the current situation for optimizing the portfolio based on some assumptions of the behavior of the market [12]. The last category, the *Meta-Learning* method, attempts to combine different categories to achieve better performance [37] [8].

Recently, a novel price prediction based strategy called Robust Median Reversion (RMR) is proposed [15]. The strategy uses the L1-median-estimate [15] on historical prices of assets to predict the future prices of assets. This is followed by an update on the portfolio using these predicted prices. However, there some factors unaccounted for such as financial crisis which cause the prices of assets to often fluctuate drastically [22]. As a result, the L1-median-estimate algorithm may fail to perform satisfactorily. In view of this, machine learning methods are applied to portfolio management in recent years.

Deep learning allows a system to automatically discover the representations needed for feature detection or classification from data [35], has had impressive performances in several areas such as image classification [38], speech recognition [40], sentiment analysis [2], machine translation [6], advertising [16] and urban design [33]. It exceeds 99% accuracy in MINST dataset for image classification and achieves 95% accuracy for speech recognition. Deep learning methods can also be applied to help investment managers to manage the portfolio by using historical market data [14] [28] or predict future price [31] [39]. However, these existing deep learning methods can only predict the stock price or obtain a efficient frontier of portfolio for a single time period. They have not been used in trading directly. In contrast, our research work is novel since we are the first to propose a strategy which combines price prediction and portfolio updating so as to output a unique portfolio vector to be directly used for multiple period trading.

## 3 Methodology

The RNN-LSTM approach proposed consists of two steps, namely, the price prediction step and portfolio updating step. In the first step, price prediction is carried out. It is a form of a regression problem in which historical stock prices are treated as a observations indexed by time. In our price prediction, the historical close price of each asset is seen as a time series. This is followed by portfolio updating in which the portfolio vector is updated based on the predicted price obtained earlier. Fig 1 gives a flow chart to describe how our strategy is carried out.

### 3.1 First Step: Price Prediction

We use RNN-LSTM to predict the close price of daily trading data. Fig 2 gives the architecture of the RNN-LSTM. The input vector of neural network is

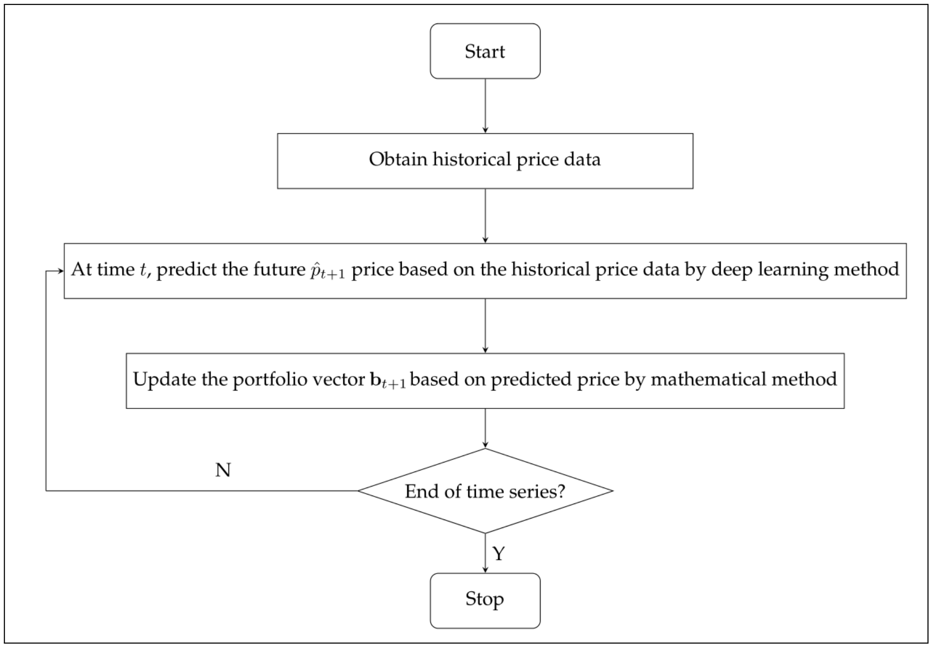


Fig. 1: A schematic description of the proposed RNN-LSTM technique. The first step of the approach predicts the future price for each single asset, and the second step updates the portfolio vector according to the predicted price.

a moving window $\mathbf{p}_t^j = \left[p_{t-i+1}^j \, p_{t-i+2}^j \, \cdots \, p_t^j\right]^T \in \mathbb{R}_+^i$ containing the most recent $i$ daily close price for the $j^{th}$ single asset. In this research, we shall set $i$ to be equal to 6 after hyper-parameter tuning. In Fig 2, the output vector $\mathbf{h}_t^j$ is such $\mathbf{h}_t^j = \left[\hat{p}_{t-i+2}^j \, \hat{p}_{t-i+3}^j \, \cdots \, \hat{p}_{t+1}^j\right]^T \in \mathbb{R}_+^i$ and the element $\hat{p}_{t+1}^j$ is the predicted close price of the $j^{th}$ single asset at time $t+1$ which be used for portfolio updating. The output window length is also set to 6.

Table 1 contains relevant information regarding the LSTM networks which is used in the training of a single asset after the parameter tuning. Therefore we have 100 neural network models for training 100 stocks. Also, to avoid the problem of over-fitting, the L1 regularization was added in our loss function and our loss function is defined as $\sum_{i=7}^{n} (\hat{y}_i - y_i)^2 + 0.01 \sum_{c=1}^{m} |\omega_c|$ where $\hat{y}_i$ is the predicted value, $y_i$ is the real value and $\sum_{c=1}^{m} |\omega_c|$ represents the sum of the absolute value of $m$ weighting parameters for the hidden layer of the neural network.

| | |
|---|---|
| **Length of window** | 6 |
| **Hidden Unit** | 8 |
| **Learning rate** | 0.0006 |
| **Batch Size** | 8 |
| **Number of iterations** | 50 |
| **Optimizer** | ADAM |
| **Period of Training Data** | 2008/8/6 - 2014/1/9 |
| **Period of cross validation Data** | 2014/1/10 - 2014/8/28 |
| **Period of Back-test Data** | 2014/8/29 - 2016/3/9 |

Table 1: The parameters of LSTM used in the experiment. These hyper-parameters are the same for each training neural network of 100 stocks in total.

### 3.2 Second Step: Portfolio Updating

In the second step, constrained optimization is used for the portfolio updating based on the predicted price $p^j_{t+1}$ for single asset $j$ obtained at time $t$. The price relative vector

$$\hat{\mathbf{x}}_{t+1} = \left[\hat{x}^1_{t+1}\ \hat{x}^2_{t+1}\ \ldots\ \hat{x}^d_{t+1}\right]^T \in \mathbb{R}^d_+$$

is a vector contains all $d$ assets at time $t+1$ whose $j^{\text{th}}$ element is $\hat{x}^j_{t+1} = \frac{\hat{p}^j_{t+1}}{p^j_t}$ in which $\hat{p}^j_{t+1}$ is the predicted close price of the $j^{\text{th}}$ single asset at time $t+1$ and $p^j_t$ is the close price of the $j^{\text{th}}$ single asset at time $t$, and the parameter $d$ is the total number of assets used in the experiment. The optimization problem to obtain the optimal portfolio can be formulated as [25]

$$\mathbf{b}_{t+1} = \begin{cases} \underset{\mathbf{b}\in\Delta_d}{\arg\min}\, \frac{1}{2}\|\mathbf{b}-\mathbf{b}_t\|^2, & \text{such that } \mathbf{b}^T \cdot \hat{\mathbf{x}}_{\mathbf{t+1}} \geq \varepsilon \quad (1) \\ \mathbf{b}_t, & \text{otherwise} \quad (2) \end{cases}$$

where $\varepsilon$ denotes the minimal return we wish to obtain in next trading day. In this research, we select $\varepsilon$ to be equal to 1.05. The vector $\mathbf{b}_t$ is the portfolio weight vector which represents the proportion of our capital which we invest in each single asset in time $t$, $\Delta_d = \left\{\mathbf{b} : b^j \geq 0, \sum\limits_{j=1}^{d} b^j = 1\right\}$ and $\|\cdot\|$ denotes the Euclidean norm. The inequality in (1) is used to decide whether we should update our portfolio vector; if its constraint is satisfied, that is, the expected return is higher than the expected minimal return ($\mathbf{b}^T \cdot \hat{\mathbf{x}} \geq \varepsilon$), then the resulting portfolio equals to previous portfolio ($\mathbf{b}_{t+1} = \mathbf{b}_t$). However, if the constraint is not satisfied, then the formulation will update a new portfolio such that the expected return is higher than the expected minimal return, while the new portfolio is not far from previous portfolio. Since the proportion which we invest in any single asset cannot be a negative number, we shall constrain our portfolio vector to be non-negative. The solution of the optimization problem is first analytically obtained without considering the

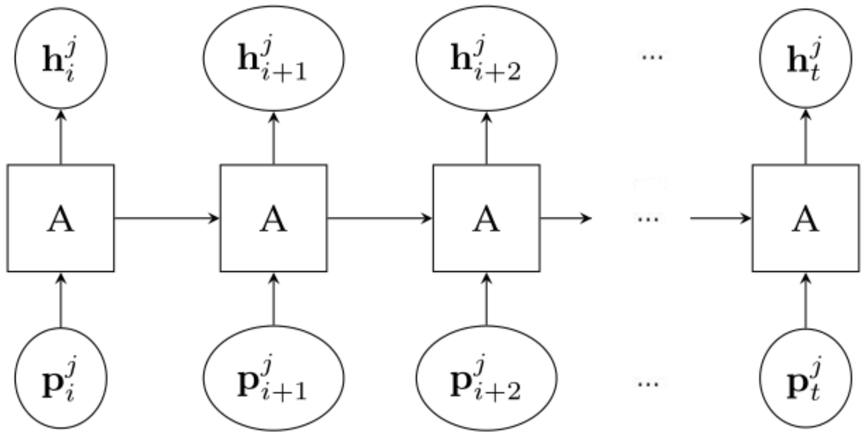


Fig. 2: The architecture of RNN used in the experiments for each single asset where $\mathbf{p}_t^j = \left[p_{t-i+1}^j \, p_{t-i+2}^j \, \dots \, p_t^j\right]^T$ is the input vector and $\mathbf{h}_t^j$ is such $\mathbf{h}_t^j = \left[\hat{p}_{t-i+2}^j \, \hat{p}_{t-i+3}^j \, \dots \, \hat{p}_{t+1}^j\right]^T$ is the output vector. Each block $A$ here represents identical hidden layers of RNN with LSTM block.

non-negatively constraint:

$$\mathbf{b}_{t+1}^{'} = \mathbf{b}_t - \alpha_{t+1} \left(\hat{\mathbf{x}}_{t+1} - \bar{x}_{t+1}\mathbf{1}\right)$$

where $\bar{x}_{t+1} = \frac{1}{d}(\mathbf{1} \cdot \hat{\mathbf{x}}_{\mathbf{t+1}})$ denotes the average predicted price relative, $d$ is the number of assets and $\alpha_{t+1}$ is the Lagrangian multiplier due to the inequality in (1) according to the method of solving the Lagrangian multiplier in a inequality constraint [27]. The Lagrangian multiplier can be calculated as

$$\alpha_{t+1} = \min \left\{ 0, \frac{\mathbf{b}_t^T \cdot \hat{\mathbf{x}}_{t+1} - \varepsilon}{\left\|\hat{\mathbf{x}}_{t+1} - \bar{x}_{t+1}\mathbf{1}\right\|^2} \right\}. \tag{3}$$

By combining (1) and (3), we can update our portfolio vector as follows:

$$\mathbf{b}_{t+1}^{'} = \mathbf{b}_t - \min \left\{ 0, \frac{\mathbf{b}_t^T \cdot \hat{\mathbf{x}}_{t+1} - \varepsilon}{\left\|\hat{\mathbf{x}}_{t+1} - \bar{x}_{t+1}\mathbf{1}\right\|^2} \right\} \left(\hat{\mathbf{x}}_{t+1} - \bar{x}_{t+1}\mathbf{1}\right). \tag{4}$$

Note that it is possible that the resulting portfolio vector $\mathbf{b}_{t+1}^{'}$ in (4) contains negative elements since the non-negativity constraint is not considered. Thus, to ensure that the portfolio is non-negative, the the resulting portfolio vector $\mathbf{b}_{t+1}^{'}$ in (4) undergoes Euclidean projection to a non-negative domain [10]. The resulting portfolio $\mathbf{b}_{t+1}$ of the projection can be proved to be the vector with the shortest Euclidean distance from vector $\mathbf{b}_{t+1}^{'}$ in non-negative domain [10].

## 4 Results and Discussion

There are 100 stocks in total from Shanghai Stock Exchange (SSE) or National Association of Securities Dealers Automated Quotations (NASDAQ) which are used in our experiments. These trading records of stocks can be downloaded in Yahoo Finance for free. The detailed information of these stocks are in the appendix. These data are divided into three parts based on time sequence, the first part is training set which is used to train a RNN-LSTM network to predict price; the second part is cross validation set which is used to tune the hyper-parameters of the neural network. The other is test set which is used in the back-test. For each stock, there are 1620 trading records used to train the neural network, 180 trading records used to hyper-parameter tuning and 360 trading records used in the back-test experiment.

### 4.1 Results of Back-tests

**Performance Measures** The following financial metrics shall be used to measure the performance of each portfolio management strategy in this paper.

1. **Final Cumulative Portfolio Wealth**. Final cumulative portfolio wealth is the portfolio value in the last time step, it can reflect how much money will be make or lost in the whole trading period. The higher the final value, the better the result of the strategy become.
2. **Positive Days**. Positive days is the proportion of trading periods which have the positive return ( $\frac{p_{t+1}}{p_t} > 1$).
3. **Max Drawdown [29]**. The drawdown is the measure of the decline from a historical peak in some variable (typically the cumulative profit or total open equity of a financial trading strategy). For example, if $X = (X(t), t \geqslant 0)$ is a random process with $X(0) = 0$, the drawdown at time $T$, denoted $D(T)$, is defined as:
$$D(T) = \max\left\{0, \max_{t \in (0,T)} X(t) - X(T)\right\}$$
The maximum drawdown (MDD) up to time $T$ is the maximum of the Drawdown over the history of the variable. The formula is:
$$M(T) = \max_{\tau \in (0,T)}\left[\max_{t \in (0,\tau)} X(t) - X(\tau)\right]$$
It can be understood as the proportion of money one will lose in the worst situation during the trading period so the lower the max drawdown, the better the result of the strategy.
4. **Sharpe Ratio [36] [32]**. In finance, the Sharpe ratio (also known as the Sharpe index, the Sharpe measure, and the reward-to-variability ratio) is a way to examine the performance of an investment by adjusting for its risk. The ratio measures the excess return (or risk premium) per unit of deviation

in an investment asset or a trading strategy, typically referred to as risk. The Sharpe ratio is defined as:

$$s_a = \frac{\mathbb{E}\left[R_a - R_f\right]}{\sigma_a} = \frac{\mathbb{E}\left[R_a - R_f\right]}{\sqrt{Var\left[R_a - R_f\right]}}$$

where $R_a$ is the asset return, $R_f$ is the risk-free return. $\mathbb{E}\left[R_a - R_f\right]$ is the expected value of the excess of the asset return over the benchmark return, and $\sigma_a$ is the standard deviation of the asset excess return. The Sharpe ratio characterizes how well the return of an asset compensates the investor for the risk taken so the higher value, the better the result of the strategy become.

**Results of Back-tests and Discussion** The initial value of the bakc-test experiment is set to be 1000000 and the commission fee is set to be 0.05% in this experiment. The commission fee is the money which will be cost in each transaction, 0.05% commission fee means 0.05% amount of transactions should be paid as commission fee in each transaction. After running the back-test experiment based on 100 stocks, the performance of RNN-LSTM based strategy is compared to several well-known or recently published strategies based on several metrics as discussed in this section. Also, we try to compare the result of our strategy to the performance of the market. Buy and Hold strategy, a strategy which spread the total capital equally into the preselected assets and holding them without making any purchases or selling until the end, can represents the performance of the market. Also, Uniform Constant Rebalanced Portfolios, a strategy invest all assets in average in the first time step and keep the capital in each assets equal in the following trading periods [21], can represents the performance of the market as well.

Most of the strategies compared in this work were surveyed by Li and Hoi [24] including Online Moving Average Reversion Strategy (OLMAR) [23], Passive Aggressive Median Reversion Strategy (PAMR) [26], Online Newton Selection (ONS) [1], Exponentiated Gradient (EG) and Anticor [4], Kernel-Based Strategy (BK) [11], Confidence Weighted Median Reversion Strategy (CWMR) [25] except Robust Median Reversion Strategy (RMR) [15]. Table 2 shows performance of 100 stocks according to the metrics Final Value, Max Drawdown, Positive Days and Sharpe Ratio of 100 stocks in the back-test with 0.05% commission fee.

Table 2 shows back-test result of 100 stocks according to the metrics Final Value, Max Drawdown, Positive Days and Sharpe Ratio of 100 stocks in the back-test with 0.05% commission fee. The value which has bold font represents the best result of these strategies, the value which has underline represents the second better result of these strategies. The final value of Buy and Hold strategy after 360 trading periods is about 100% which shows the overall trend of the market is stable relatively. It can be seen that RNN-LSTM strategy has the highest return (148%) and CWMR strategy obtains the second best return(124%) in the back-test experiment. RNN-LSTM strategy also achieves the best result in Sharpe Ratio and achieves the second best result in Max Drawdown. RNN-LSTM strategy outperforms than RMR strategy, which is the benchmark in this

| | Final value | Max Drawdown | Sharpe Ratio | Positive Days |
|---|---|---|---|---|
| ANTICOR | 106.2414% | 0.234585 | 0.409714 | 0.51676 |
| BAH | 99.9219% | 0.199235 | 0.120446 | 0.511173 |
| BK | 103.0641% | 0.147922 | 0.31184 | 0.541899 |
| CRP | 107.7313% | 0.165124 | 0.96716 | 0.513966 |
| CWMR | <u>124.6668%</u> | 0.21815 | <u>1.417289</u> | <u>0.555866</u> |
| EG | 116.6944% | **0.125781** | 1.237309 | 0.519553 |
| OLMAR | 101.4545% | 0.415075 | 0.08132 | 0.485101 |
| ONS | 86.86563% | 0.296049 | -0.7179 | 0.472067 |
| PAMR | 120.174% | 0.227172 | 1.390486 | **0.558659** |
| LSTM | **148.4522%** | <u>0.139873</u> | **1.687191** | 0.527933 |
| RMR | 111.3425% | 0.178053 | 0.780039 | 0.5 |
| UP | 108.4251% | 0.141192 | 0.930596 | 0.519553 |

Table 2: Performance of our strategy and other strategies in back-test of all 100 stocks with 0.05% commission fee. The performance metrics are Final Portfolio Value, Max Drawdown, Sharpe Ratio. The other strategies in the table are Buy and Hold (BAH), Uniform Constant Rebalanced portfolio (CRP) [21], Online Moving Average Reversion Strategy (OLMAR) [23], Robust Median Reversion Strategy (RMR) [15], Passive Aggressive Median Reversion Strategy (PAMR) [26], Online Newton Selection (ONS) [1], Exponentiated Gradient (EG), Kernel-Based Strategy (BK) [11], Confidence Weighted Median Reversion Strategy (CWMR) [25] and Anticor [4]. The value which has bold font represents the best result of these strategies, the value which has underline represents the second best result of these strategies.

research, in all four metrics. RNN-LSTM strategy does not achieve the best two result only in the Postive Days metric. It means the stability of RNN-LSTM strategy may not be remarkable enough.

Figure 3 gives the plot of the change of final value against time of the RNN-LSTM strategy, Buy and Hold strategy and Uniform Constant Rebalanced Portfolios. The Buy and Hold strategy and Uniform Constant Rebalanced Portfolios can represent the performance of the whole market. The $x$-axis represents the time $t$ and the $y$-axis represents the final value of the each strategy. RNN-LSTM strategy performs better than Buy and Hold strategy and UCRP strategy in final value throughout the back-tests although it cannot perform better in every time period. It is because the profitability of our strategy depends on the accuracy of the price prediction. The price prediction can not be accurate every time and the money will be lose if the prediction is not accurate enough in that trading period. However, RNN-LSTM strategy still obtains a lower value in Max Drawdown than Buy and Hold strategy or

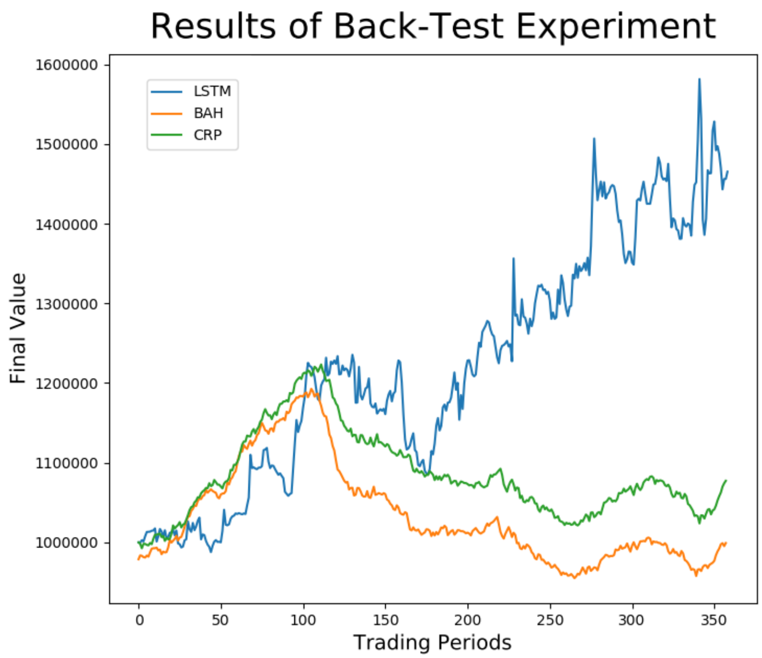


Fig. 3: Final value of the Back-test for RNN-LSTM strategy Buy, Hold strategy and Uniform Constant Rebalanced Portfolios in each trading day.

Uniform Constant Rebalanced Portfolios which means that the risk of our strategy is lower than the average performance of the market in terms of the whole 360 trading periods.

## 5 Conclusions and Future Work

In this paper, we propose a novel multiple period on-line portfolio selection strategy which based on the stock price prediction by Recurrent Neural Network (RNN) which has Long Short-Term Memory (LSTM) block. The profitability of our strategy surpasses most of common portfolio management strategies, as demonstrated in the paper by the average result of back-test over 100 stocks in a stock market. In the experiment, RNN-LSTM strategy outperformed RMR strategy, which is seen as benchmark in this research for all four metrics. The satisfying performance of our strategy confirm the effectiveness of prediction of RNN-LSTM. Also, our strategy may can initiate a new direction of portfolio management research which combines deep learning method to constraint optimization method.

There are several research directions we may take in the future. The first is multiple comparisons with the best (MCB) [19]. We can divide these stocks into several small datasets and MCB method may be used in these datasets to obtain a more compelling conclusion in comparison with our strategies and other strategies. Ranking test can reveal that whether the excellent performance of our strategy is not due to chance but owed to the strategy principle. Next, we shall try to upgrade our strategy to control the volatility of the portfolio. Our strategy do not achieve the best result of Max drawdown and Positive Days in the back-test experiment which means the stability of our strategy is not remarkable enough. In additional, we can put risk-free asset such as T-bills into consideration to avoid the risk which all prices of stocks are falling down. The volatility of portfolio should also be controlled when risk-free asset is put into consideration. Lastly, we shall continue to explore the effectiveness of our strategy in high frequency trading data such as half hour trading data or 5 minutes trading data and compared our strategy with other machine learning strategy such as the deep reinforcement learning strategy [17] in the future work.

## References


1. Amit Agarwal, Elad Hazan, Satyen Kale, and Robert E. Schapire. Algorithms for portfolio management based on the newton method. In *International Conference*, pages 9–16, 2006.
2. Oscar Araque, Ignacio Corcuera-Platas, J. Fernando Sanchez-Rada, and Carlos A. Iglesias. Enhancing deep learning sentiment analysis with ensemble techniques in social applications. *Expert Systems with Applications An International Journal*, 77(C):236–246, 2017.
3. Michael J. Best and Jaroslava Hlouskova. The efficient frontier for bounded assets. *Mathematical Methods of Operations Research*, 52(2):195–212, 2000.
4. A. Borodin, R. Elyaniv, and V. Gogan. Can we learn to beat the best stock. *Journal of Artificial Intelligence Research*, 21(1):579–594, 2011.
5. Robert G. Edgett Cooper. New product portfolio management:. *Journal of Product Innovation Management*, 16(4):333–351, 1999.
6. Marta R. Costa-jussa, Alexandre Allauzen, Loic Barrault, Kyunghun Cho, and Holger Schwenk. Introduction to the special issue on deep learning approaches for machine translation. *Computer Speech & Language*, 46, 2017.
7. T. M. Cover. Universal data compression and portfolio selection. In *Symposium on Foundations of Computer Science*, page 534, 1996.
8. Puja Das and Arindam Banerjee. Meta optimization and its application to portfolio selection. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1163–1171, 2011.
9. Davelos, L Anita, Kinkel, L Linda, Samac, and A Deborah. *Modern investment theory*. Prentice Hall,, 2001.
10. John C. Duchi, Shai Shalev-Shwartz, Yoram Singer, and Tushar Chandra. Efficient projections onto the. In *International Conference*, pages 272–279, 2008.
11. Gyorfi, Lugosi, Gabor, Udina, and Frederic. Nonparametric kernel-based sequential investment strategies. *Mathematical Finance*, 16(2):337–357, 2010.
12. Laszlo Gyorfi, Gabor Lugosi, and Frederic Udina. Nonparametric kernel-based sequential investment strategies. *Mathematical Finance*, 16(2):337C357, 2006.
13. N. Ozsoylev Han and Shino Takayama. Price, trade size, and information revelation in multi-period securities markets. *Journal of Financial Markets*, 13(1):49–76, 2010.

14. J. B. Heaton, N. G. Polson, and J. H. Witte. Deep learning for finance: deep portfolios. *Applied Stochastic Models in Business and Industry*, 33(1), 2017.
15. Dingjiang Huang, Junlong Zhou, Bin Li, Steven C. H. Hoi, and Shuigeng Zhou. Robust median reversion strategy for on-line portfolio selection. In *International Joint Conference on Artificial Intelligence*, volume 28, pages 2006–2012, 2013.
16. Hsien De Huang, Chia Mu Yu, and Hung Yu Kao. Data-driven and deep learning methodology for deceptive advertising and phone scams detection. 2017.
17. Zhengyao Jiang and Jinjun Liang. Cryptocurrency portfolio management with deep reinforcement learning. *Arxiv*, 2017.
18. Kelly, L J., and Jr. A new interpretation of information rate. *Bell System Technology Journal*, 2(3):917–926, 1956.
19. Alex J. Koning, Philip Hans Franses, Michele Hibon, and H. O. Stekler. The m3 competition: Statistical tests of the results. *International Journal of Forecasting*, 21(3):397–409, 2005.
20. John R. Koza. Genetic programming as a means for programming computers by natural selection. *Statistics & Computing*, 4(2):87–112, 1994.
21. Suleyman S. Kozat and Andrew C. Singer. Universal constant rebalanced portfolios with switching. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages III–1129 – III–1132, 2007.
22. Jan Lansky. Analysis of cryptocurrencies price development. *Acta Informatica Pragensia*, 2016.
23. Bin Li and Steven C. H. Hoi. On-line portfolio selection with moving average reversion. *Papers*, page 173C190, 2012.
24. Bin Li and Steven C. H. Hoi. Online portfolio selection: A survey. *Acm Computing Surveys*, 46(3):1–36, 2012.
25. Bin Li, Steven C. H. Hoi, Peilin Zhao, and Vivekanand Gopalkrishnan. Confidence weighted mean reversion strategy for online portfolio selection. *Acm Transactions on Knowledge Discovery from Data*, 7(1):1–38, 2013.
26. Bin Li, Peilin Zhao, Steven C. Hoi, and Vivekanand Gopalkrishnan. Pamr: Passive aggressive mean reversion strategy for portfolio selection. *Machine Learning*, 87(2):221–258, 2012.
27. Mengmou Li. Generalized lagrange multiplier method and kkt conditions with application to distributed optimization. *IEEE Transactions on Circuits & Systems II Express Briefs*, PP(99):1–1, 2018.
28. Shanghong Li, Jiayu Zhang, and Yan Qi. Predicting s & p500 index using artificial neural network. In *International Conference on Computer and Computing Technologies in Agriculture*, pages 173–189, 2015.
29. Zhiyong Li. Maxdrawdown: Stata module to calculate the maximum drawdown of a stock, fund or other financial product. *Statistical Software Components*, 2016.
30. Harry Markowitz. Portfolio selection. *Journal of Finance*, 7(1):77–91, 1952.
31. Phayung Meesad and Srikhacha Tong. Stock price time series prediction using neuro-fuzzy with support vector guideline system. In *Acis International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/distributed Computing*, pages 422–427, 2008.
32. Christoph Memmel. Performance hypothesis testing with the sharpe ratio. *Social Science Electronic Publishing*, 27(3):299–306, 2003.
33. Vahid Moosavi. Urban morphology meets deep learning: Exploring urban forms in one million cities, town and villages across the planet. 2017.
34. Maryam M Najafabadi, Flavio Villanustre, Taghi M Khoshgoftaar, Naeem Seliya, Randall Wald, and Edin Muharemagic. Deep learning applications and challenges in big data analytics. *Journal of Big Data*, 2(1):1, 2015.

35. Foundations Trends In Signal Processing. Deep learning: Methods and applications. *Foundations & Trends in Signal Processing*, 7(3):197–387, 2014.
36. William F. Sharpe. The sharpe ratio. *Journal of Portfolio Management*, 21(1):49–58, 1994.
37. V. Vovk and C. Watkins. Universal portfolio selection. In *Eleventh Conference on Computational Learning Theory*, pages 12–23, 1998.
38. Jiajun Wu, Yinan Yu, Chang Huang, and Kai Yu. Deep multiple instance learning for image classification and auto-annotation. In *Computer Vision and Pattern Recognition*, pages 3460–3469, 2015.
39. Yoshiyuki Yabuuchi and Junzo Watada. *Building Fuzzy Autocorrelation Model and Its Application to the Analysis of Stock Price Time-Series Data.* Springer Berlin Heidelberg, 2013.
40. Dong Yu and Li Deng. *Automatic Speech Recognition: A Deep Learning Approach.* Springer, 2014.

# 6 Appendix

The appendix shows the detailed stock code or logogram of stocks which are used in the back-test.

1. **SSE**. There are 50 stocks from SSE, their stock codes are 600000, 600001, 600004, 600015, 600028, 600031, 600060, 600249, 600546, 600848,600104, 600109, 600119, 600485, 600893, 601198, 601377, 601800, 601985, 601998,600016, 600036, 600111, 600519, 600585, 601006, 601088, 601318, 601328, 601601,600048, 600050, 600089, 600104, 600282, 600348, 600547, 601857, 601899, 601939,600019, 600362, 600383, 600489, 600518, 600887, 601600, 601628, 601788, 601766.
2. **NASDAQ**. There are 50 stocks from NASDAQ, their logograms are CAT, GE, GS, F, CAH, CCL, CCE, DIS, DUK, HAS, AVP, BXP, D, DFS, DVA, IFF, MAS, MO, POM, USB, ALL, BDX, C, CNP, EFX, MSI, NWL, S, TGNA, ZMH, BBT, BBY, BIG, BILL, COP, DRI, GWW, VNO, XEL, XL, AEP, AIV, AN, BMY, CHRW, CL, DNR, HUM, JPM, MTB.