

Article

# Portfolio Optimization-Based Stock Prediction Using Long-Short Term Memory Network in Quantitative Trading

Van-Dai Ta , CHUAN-MING Liu \*  and Direselign Addis Tadesse 

Department of Computer Science and Information Engineering, National Taipei University of Technology, Taipei 106, Taiwan; t104999002@ntut.edu.tw (V.-D.T.); t106999405@ntut.edu.tw (D.A.T.)

\* Correspondence: cmliu@ntut.edu.tw; Tel.: +886-2-2771-2171

Received: 1 December 2019; Accepted: 4 January 2020; Published: 7 January 2020



**Abstract:** In quantitative trading, stock prediction plays an important role in developing an effective trading strategy to achieve a substantial return. Prediction outcomes also are the prerequisites for active portfolio construction and optimization. However, the stock prediction is a challenging task because of the diversified factors involved such as uncertainty and instability. Most of the previous research focuses on analyzing financial historical data based on statistical techniques, which is known as a type of time series analysis with limited achievements. Recently, deep learning techniques, specifically recurrent neural network (RNN), has been designed to work with sequence prediction. In this paper, a long short-term memory (LSTM) network, which is a special kind of RNN, is proposed to predict stock movement based on historical data. In order to construct an efficient portfolio, multiple portfolio optimization techniques, including equal-weighted modeling (EQ), simulation modeling Monte Carlo simulation (MCS), and optimization modeling mean variant optimization (MVO), are used to improve the portfolio performance. The results showed that our proposed LSTM prediction model works efficiently by obtaining high accuracy from stock prediction. The constructed portfolios based on the LSTM prediction model outperformed other constructed portfolios-based prediction models such as linear regression and support vector machine. In addition, optimization techniques showed a significant improvement in the return and Sharpe ratio of the constructed portfolios. Furthermore, our constructed portfolios beat the benchmark Standard and Poor 500 (S&P 500) index in both active returns and Sharpe ratios.

**Keywords:** stock prediction; LSTM; portfolio optimization; quantitative trading

---

## 1. Introduction

A portfolio is defined as a collection of investment assets. Portfolio management refers to the process of investment decision making based on customized tactical investment strategies to match maximize the return for each investing time horizon. There are two popular approaches to manage the investment portfolio: traditional and quantitative. Both approaches share some common characteristics such as investigating a small set of key-driving factors of equity values, analyzing historical data to estimate these key drivers, adopting eligibility criteria for stock-selection decisions, and evaluating the performance over time. However, while traditional portfolio management relies heavily on the judgment depth analysis, regime shifts, key characteristics, and qualitative factors, quantitative portfolio management focuses on universe exploration, discipline, verification, risk management, and lower fees. Not only can it uncover more opportunities, but it can also do a better job of controlling unintended risks [1].

Quantitative trading consists of trading strategies based on quantitative investment analysis, which relies on mathematical models to design an automated trading system. In quantitative trading,

portfolio construction is the process of selecting and allocating investment on multiple stocks, which can be understood as diversification in quantitative trading in order to minimize the risk in trading. Market trend, entry and exit trade, price history, and volume are the key factors for each quantitative trading strategy. Developing an accurate forecasting model is considered as the most critical process to construct an efficient portfolio in the quantitative approach. In quantitative trading, stock prediction plays an important role in forecasting the movement of the market in general or a particular stock. Forecasting the stock price has been considered as one of the most challenging tasks in the financial market owing to the complexity of multivariate time series attributes as well as the amount of involved financial data. Numerous studies have been carried out to enhance prediction accuracy such as statistical and machine learning approaches [2]. Recently, artificial intelligence (AI) and deep learning algorithms offer a number of potential advantages over existing traditional prediction models on both accuracy and decision-making support. Deep learning algorithms allow for designing multiple trading strategies that are implemented consistently and are able to adapt to a real-time market [3,4]. Although deep learning has been extensively studied for its potentials in stock prediction, little attention has been paid to take advantage of the stock prediction phase to construct efficient quantitative portfolios. In this paper, a special variation of recurrent neural network (RNN), long short-term memory (LSTM), is proposed to build a prediction model for the stock price prediction, and then portfolio optimization techniques are applied to leverage the prediction results. Multiple quantitative portfolios are constructed based on a strategic asset allocation trading strategy. For each experiment, the prediction model achieves high accuracy in prediction, and our constructed portfolios have a considerable return in multiple predicted time periods compared with actual trading. The constructed portfolios outperform to the benchmark Standard and Poor 500 (S&P 500) index in terms of active return and risk control. The main contributions of this paper are summarized as follows:

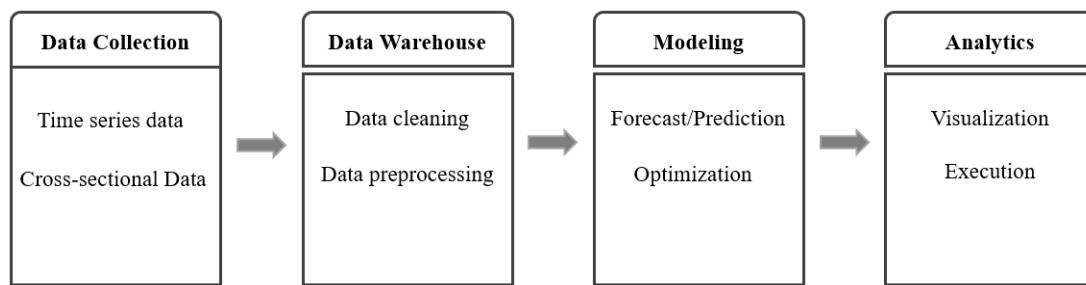
- The LSTM prediction model was proposed to predict stock price in order to construct and optimize portfolios in quantitative trading.
- Presenting a comparison between LSTM prediction model performance to gated recurrent units (GRUs) and other conventional machine learning models such as linear regression (LR) and support vector regression (SVR) for stock prediction.
- Simulation modeling and optimization modeling approaches were used to optimize portfolios in quantitative trading.
- Finally, portfolio performance evaluation for the constructed portfolios was conducted in which our constructed portfolios outperform the benchmark on both active return and risk control.

The remaining part of the paper is structured as follows. The basic concepts in quantitative trading and related work are presented in Section 2. The proposed LSTM prediction models for stock prediction and portfolio optimization techniques are discussed in Section 3. The experiment and results are presented in Section 4. Finally, conclusions and discussions are summarized in Section 5.

## 2. Background and Literature Review

### 2.1. Fundamentals of Quantitative Trading

A typical system for quantitative investment management is outlined in Figure 1. The first fundamental piece of the system is the data collection process, in which data can be gathered from external sources, from a data vendor, or from proprietary research. Generally, there are two types of financial data structures: time-series data and cross-sectional data [5]. Data cleaning and preprocessing are the main tasks in order to get reliable data sources stored in the data warehouse. The role of the modeling process mainly focuses on building accurate prediction, statistical analysis, and optimization models. Finally, the results of the analysis are visualized and become the criteria for investment decision-making. The last two stages: modeling and analytics, are often employed in an iterative process of evaluating trends, determining strategies, backtesting, and assessing portfolio performance.



**Figure 1.** A type of quantitative investment management system.

Quantitative trading is an automated trading system in which the trading strategies and decisions are conducted by a set of mathematical models. The idea of quantitative trading is designed to leverage statistical mathematics, computer algorithms, and computational resources for high-frequency trading systems, which aims to minimize risk and maximize return based on the historical performance of the encode strategies tested against historical financial data. In quantitative portfolio management, quantitative trading is considered as the new era of trading that provides investors a variety of benefits from efficient execution to less transaction costs, as well as take advantage of technical tactics to improve portfolio performance. As the advance of computational resources, trading systems are required to digest massive financial data under various formats and quickly reacted to the changing of market conditions. Quantitative trading is extremely well suited for a high-frequency trading system. It became popular in the early 2000s. By 2005, it accounted for about 25% of the total volume. The industry faced an acceleration of quantitative trading, where volumes increased threefold to 75% in 2009. Quantitative trading also provides investors with many benefits such as lower commissions, anonymity, control, discipline, transparency, access, competition, and reduced transaction costs [6]. A typical quantitative trading system has five modules: alpha model, risk model, transaction model, portfolio construction model, and execution model. Quantitative trading strategy workflow consists of six stages: data collection, data preprocessing, trade analysis, portfolio construction, back-testing, and execution [7].

## 2.2. Quantitative Portfolio Management

### 2.2.1. Portfolio Construction

Portfolio construction attempts to construct an efficient portfolio that maximizes expected return for a given level of risk, or equivalently, minimizes the risk for a given expected return on a specific investment time horizon. In general, portfolio construction is the decision making about asset allocation and security selection. Asset allocation is often used to describe the money management strategy that designates how capital should be distributed into various asset classes, or broad types of investments such as stocks, bonds, commodities, and cash within an investment portfolio. Most asset allocation techniques fall within six distinct strategies: strategic asset allocation, tactical asset allocation, dynamic asset allocation, constant-weight asset allocation, insured asset allocation, and integrated asset allocation [8]. On the basis of investment strategy, risk tolerance, and liability utilization, the portfolio construction strategy can be classified as either active or passive portfolio. Security selection is the process of identifying individual securities within a certain asset class that will make up the portfolio. Security selection comes after the asset allocation has been set. After the asset allocation strategy has been developed, securities must be selected to construct the portfolio and populate the allocation targets according to the strategy. While asset allocation is based on investing strategies, security selection heavily relies on a prediction or forecast. Hence, a precise investing strategy making sure a portfolio has the right mix of assets to suit individual circumstances, investment objectives, and attitude to risk with the highly accurate prediction model is the key to determine the expected portfolio return. There are three key inputs for portfolio construction: expected return, variance of asset returns (volatility), and correlation (or covariance) of asset returns. The expected return of a portfolio provides

an estimate of how much return one can get from a portfolio. The variance gives an estimate of the risk that an investor is taking while holding that portfolio. The returns and the risk of the portfolio depend on the returns and risks of the individual stocks and their corresponding shares in the portfolio.

Quantitative portfolio risk management often relies on statistical measures related to the spread or the tails or distribution of portfolio returns. Such measures include variance and standard deviation (spread), coefficient of variation (risk relative to mean), and percentiles of the distribution (tails). The concept of risk in finance investment is captured in many ways. However, the basic and most widely used one is concerned with risk as an uncertain variable that will fall from what one expects. Therefore, a natural way to define a measure of uncertainty is as the average spread or dispersion of a distribution. There are two aspects of risk: the distances between possible values and the expectation, and the probabilities of attaining the various possible values. Two measures that describe the spread of the distribution are variance and standard deviation, in which the standard deviation is the square root of the variance. The higher spread or dispersion indicates a higher variance/standard deviation that could be considered a higher risk.

The idea behind covariance is to measure simultaneous deviations from the means for two random variables. The problem with covariance is that its units are products of the original units or the two random variables, so the value for covariance is difficult to interpret. The correlation coefficient divides the covariance by the product of the standard deviations of the two random variables.

## 2.2.2. Portfolio Optimization

In general, portfolio optimization techniques are proposed to optimal asset allocation in order to maximize a portfolio return and minimize its risk. Modern portfolio theory was a theory on how risk-averse investors can construct portfolios to optimize or maximize expected return based on a given level of risk, emphasizing that risk is always an inherent part of higher reward [9]. Sharpe further introduced the industry to the capital asset pricing model (CAPM), which in the simplest forms, was a technique to combine the market portfolio with a risk-free asset to further improve the set of risk-return above the efficient frontier [10]. Modern portfolio theory and capital market theory provide a framework to specify and measure investment risk and to develop relationships between expected return and risk. These relationships are called asset pricing models. The arbitrage pricing theory (APT) as an alternative to the CAPM was developed in the work of [11]. Unlike in the CAPM, markets were assumed as perfectly efficient; APT was a multi-factor asset pricing model based on the idea that an asset's returns can be predicted using the linear relationship between the asset's expected return and a number of macroeconomic variables that capture systematic risk. The Fama French three-factor model was an asset pricing model that expands on the CAPM by adding size risk and value risk factors to the market risk factors. This model considers the fact that value and small-cap stocks outperform markets on a regular basis. By including these two additional factors, the model adjusts for this outperforming tendency, which is thought to make it a better tool for evaluating manager performance [12]. The Black–Litterman model was essentially a combination of two main portfolio theories: the CAPM and the modern portfolio theory [13]. The main benefit of the Black–Litterman model is that it allows the portfolio manager to use it as a tool for producing a set of expected returns within the mean-variance optimization framework. In addition to developing portfolio theories as the principle of portfolio management, multiple optimization techniques have been proposed to extend the impact of modern portfolio theory. A 60-year review of different approaches developed to address the challenges encountered when using portfolio optimization in practice, such as the transaction costs, portfolio constraints, and estimates errors was provided in [14]. Mathematical optimization has also attracted widespread interest in multi-objective optimization. There exist a whole series of optimization algorithms such as convex programming, integer programming, linear programming, and stochastic programming developed to solve optimization problems not only for linear constraints but also for random constraints [15,16]. Metaheuristic is a subfield of computational intelligence that represents an efficient way to deal with complex optimization problems and is applicable to both continuous and

combinatorial optimization problems. Evolutionary algorithms such as genetic algorithms have shown an effective impact on complex objectives and constraint optimization tasks [17]. Much research in recent years has focused on uncertainty in financial investment. Probabilistic programming techniques have also been applied to handle the uncertainty of the financial markets to support portfolio selection. The fuzzy set theory has been widely used to solve many practical problems, including financial risk management. Using fuzzy approaches, quantitative analysis, qualitative analysis, experts' knowledge, and investors' subjective strategies can be better integrated into a portfolio selection model [18]. One significant difference between the discussed approaches and this work is that the input values such as expected returns and risk for optimization models, which either are calculated by a mathematical or statistical model, are based on historical data. In quantitative trading, the expected return and risk are calculated by the alpha and risk models, respectively. In other words, input values for the optimization model are calculated based on the prediction model. Optimization is performed on predicted data, which is an important requirement for active portfolio management in quantitative trading, where dynamic and large-scale portfolio optimization is the top priority.

### 2.2.3. Portfolio Performance Evaluation

Portfolio performance evaluation is taken to test the notion of market efficiency. The evaluation process is conducted for three important benefits: increase efficiency, monitor risk, and analyze returns. There are a variety of different measures that can be used to evaluate portfolio performance. The ability to derive above-average returns for a given risk class and the ability to diversify the portfolio completely to eliminate all unsystematic risk, relative to the portfolio's benchmark, are two desirable attributes for an efficient portfolio. The performance evaluation methods generally fall into two categories, namely conventional and risk-adjusted methods [19]. The most widely used conventional methods include benchmark comparison and style comparison. The risk-adjusted methods adjust returns in order to take account of differences in risk levels between the managed portfolio and the benchmark portfolio. The risk-adjusted methods are preferred to conventional methods. Some of the most common metrics of portfolio performance are listed in the work of [20].

## 2.3. Deep Learning in Stock Prediction

With the enormous growth of financial data in volume and complexity, machine-learning algorithms provide powerful tools to extract patterns from data processed all across the global. For many years, stock prediction always has drawn attention to the development of intelligent trading systems. There are substantial benefits to be gained from stock prediction for security selection and quantitative investment analysis.

In practice, stock prediction can be conducted by fundamental analysis, technical analysis, and sentiment analysis. Fundamental analysis is the most conventional use, which tries to determine a stock's value or price based on financial statements such as income statement, balance sheet, and cash-flow statement. In other words, the main objective of fundamental analysis is to estimate a company's intrinsic value. Fundamental signals have a positive and significant correlation with future earnings performance [21]. Fundamental analysis is the prerequisite investigation for value investing as known as long-term investing. In contrast, technical analysis typically begins with charts and technical indicators based on historical data. Technical analysis is usually used to predict short- to medium-term time horizons. An artificial neural network-based stock trading system using technical analysis and big data framework has been proposed in the work of [22]. The results have shown that, by choosing the most appropriate technical indicators, the neural network model can obtain comparable results against the buy and hold strategy in most of the cases. Furthermore, fine-tuning the technical indicators and/or optimization strategy can enhance the overall trading performance.

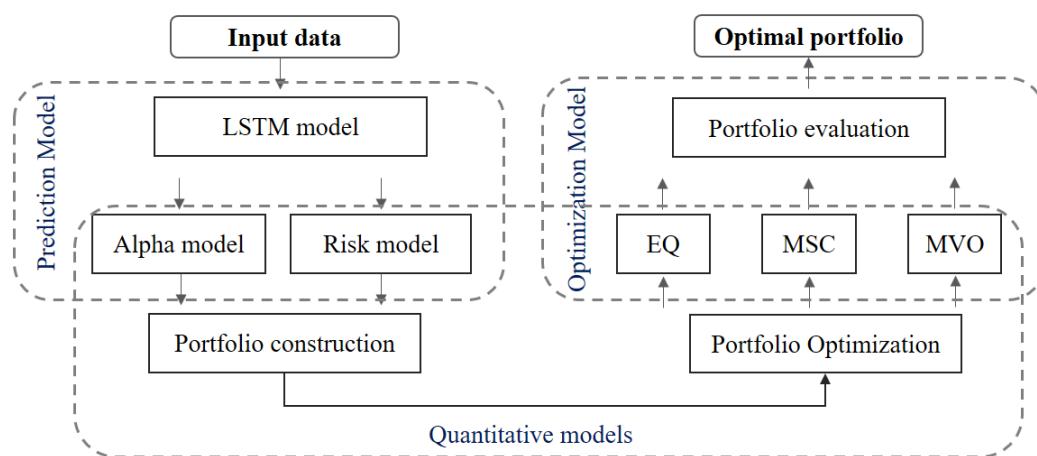
In the short-term, the stock market is irrational movement by the effect of emotion trading. Sentiment analysis is the new trend for stock prediction based on finding the correlation between public sentiment and market sentiment. The results show that social media content can give an impact

on stock price via sentiment analysis [23,24]. On the effort of improving the prediction accuracy, many studies have been conducted by combining multiple analysis approaches [25,26].

Recently, there is considerable interest in stock prediction using deep learning methods. Deep learning techniques have been receiving a lot of attention lately, with breakthroughs in image processing and natural language processing. However, its application to finance does not yet seem to be commonplace. It has been used for limit order book modeling, financial sentiment analysis, volatility prediction, and portfolio optimization [27–30]. With the effort to decompose and eliminate the noise of the stock price time-series data, the wavelet transform was used. Features are extracted from the decomposed data using stacked autoencoders, and then the high-level de-noising features are fed into long short-term memory (LSTM) to build the model and forecast the next day's closing price [31]. Stock price exchange rates are forecasted by improving the deep belief network (DBN). The structure of the DBN is optimally determined through experiments and, to accelerate the speed of learning rate, conjugate gradient methods are applied. The model shows more efficiency at foreign exchange rate prediction compared with the feedforward network (FFNN) [32]. In the work of [33], the recurrent neural network was introduced and used, however, it suffers from the vanishing gradient problem. The vanishing gradient problem was improved in the LSTM and GRU model. The LSTM model has update, input, forget, and output gates, and maintains the internal memory state and applies a non-linearity(sigmoid) before the output gate, whereas GRU has only update and reset gates.

### 3. Methodology

Our proposed methodology architecture is developed based on the typical quantitative investment management system mentioned in Section 1. Historical data and cross-sectional data were collected from multiple resources in various formats. It could be technical, fundamental, macro-economics, and sentiment data. Multiple prediction models were conducted to predict stock prices such as LR, SVR, and LSTM. On the basis of the predicted results for each period, the expected return and volatility were calculated by the alpha model and risk model, respectively. The portfolio was constructed by selecting the outperform stocks from the predicted result in terms of the highest expected return and lowest risk. Optimal stock allocation for the constructed portfolio was evaluated by simulation and optimization modeling. Equal-weights allocation (EQ), simulation modeling Monte Carlo simulation (MCS), and mean-variance optimization (MVO) were used to evaluate the optimal stock allocation weights. The overview of the architecture is presented in Figure 2.



**Figure 2.** The overview of architecture.

#### 3.1. Prediction Model

In this section, we proposed LSTM network to predict the stock price and construct the portfolio based on the prediction outputs.

LSTM network is a variant of RNN, which has memory blocks (cells) in the hidden layer that are recurrently connected. There are two states that are being transferred to the next cell: the cell state and the hidden state. The memory blocks are responsible for remembering things and manipulations to this memory are done through three major mechanisms, called gates. A forget gate is responsible for removing information from the cell state. The input gate is responsible for the addition of information to the cell state. The output gate decides which next hidden state should be selected. Operations performed on LSTM network units are explained in (1–6), where  $x_t$  is the input at time  $t$  and  $f_t$  is the forget gate at time  $t$ , which clears information from the memory cell when needed and keeps a record of the previous frame whose information needs to be cleared from the memory. The output gate  $o_t$  keeps the information about the upcoming step, where  $g$  is the recurrent unit, having activation function “ $\tanh$ ”, and is computed from the input of the current frame and state of the previous frame  $h_{t-1}$ . In all input ( $I_t$ ), forget ( $f_t$ ), and output ( $O_t$ ) gates, as well as the recurrent unit ( $g_t$ ), we use  $(W_i, W_f, W_o, W_g)$  and  $(b_i, b_f, b_o, b_g)$  as weights and bias, respectively. The input gate determines what parts of the transformed input  $g_t$  need to be added to the long-term state  $c_t$ . This process updates the long-term state  $c_t$ , which is directly transmitted to the next cell. Finally, the output gate transforms the updated long-term state  $c_t$  through  $\tanh(.)$ ; filters it by  $o_t$ ; and produces the output  $y_t$ , which is also sent to the next cell as the short-term state  $h_t$ .

The equations for LSTM computations are given by the following:

$$i_t = \sigma(W_{x_i}^T x_t + W_{h_i}^T h_{t-1} + b_i), \quad (1)$$

$$f_t = \sigma(W_{x_f}^T x_t + W_{h_f}^T h_{t-1} + b_f), \quad (2)$$

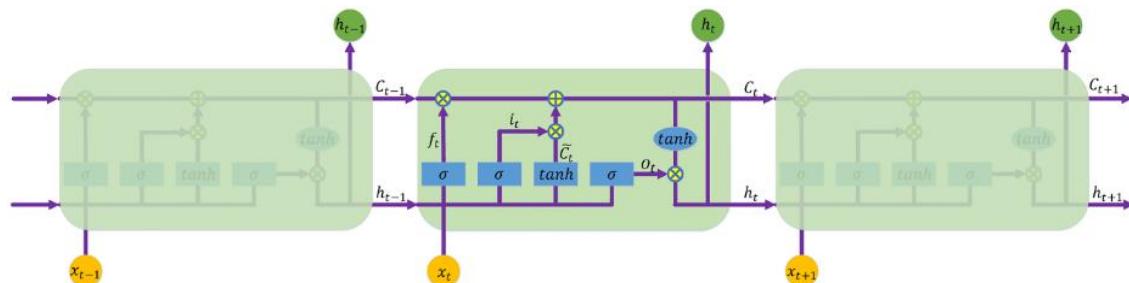
$$o_t = \sigma(W_{x_O}^T x_t + W_{h_O}^T h_{t-1} + b_O), \quad (3)$$

$$g_t = \tanh(W_{x_g}^T x_t + W_{h_g}^T h_{t-1} + b_g), \quad (4)$$

$$c_t = f_t c_{t-1} + i_t g_t, \quad (5)$$

$$y_t = h_t = o_t \tanh c_t, \quad (6)$$

where  $\sigma(.)$  is the logistic function, and  $\tanh(.)$  is the hyperbolic tangent function. The three gates open and close according to the value of the gate controllers  $f_t$ ,  $i_t$ , and  $o_t$ , all of which are fully connected layers of neurons. The range of their outputs is  $[0, 1]$ , as they use the logistic function for activation. In each gate, their outputs are fed into element-wise multiplication operations, so, if the output is close to 0, the gate is narrowed and less memory is stored in  $c_t$ , while if the output is close to 1, the gate is more widely open, letting more memory flow through the gate. Given LSTM cells, it is common to stack multiple layers of the cells to make the model deeper to be able to capture the nonlinearity of the data. Figure 3 illustrates how computation is carried out in an LSTM cell. To keep the wealth of a stock market, we have to have an efficient prediction model that can predict based on the previous data generated from the stock market. In this paper, we used LSTM networks to build a model that can predict the stock price [34,35]. On the basis of the output of the forecasted price, a portfolio is constructed.



**Figure 3.** A recurrent neural network with LSTM network architecture.

### 3.2. Quantitative Models

#### 3.2.1. Multiple Assets Portfolio Construction

Suppose that a portfolio consists of  $N$  stocks, and  $S_0$  is the set of initial value for each stock in the portfolio, denoted as  $S_0 = (s_1^0, \dots, s_N^0)$ . The number of each stock in the portfolio is denoted as  $X = (x_1, \dots, x_N)$ . The initial value of the portfolio  $V_0$  is calculated as follows:

$$V_0 = x_1 s_1^0 + \dots + x_N s_N^0 = \sum_{i=1}^N x_i s_i^0 \quad (7)$$

The decision on the number of shares in each asset will follow the decision on the division of our capital, which is our primary concern, and is expressed as the weights  $W = (w_1, \dots, w_N)$  with the constraint  $\sum_{i=1}^N w_i = 1$ , defined by  $w_i = \frac{x_i s_i^0}{V_0}$  with  $i = 1, \dots, N$ .

At the end of the period  $t$ , the values of the stocks change  $S_t = (s_1^t, \dots, s_N^t)$ , which gives the final value of the portfolio  $V_t$  as a random variable,

$$V_t = x_1 s_1^t + \dots + x_N s_N^t = \sum_{i=1}^N x_i s_i^t \quad (8)$$

The actual return of a portfolio  $R_P = (r_1, \dots, r_N)$  is the set of random returns on each stock of the portfolio, and the vector of expected return by  $\mu = (\mu_1, \mu_2, \dots, \mu_N)$  with  $\mu_i = E(r_i)$  for  $i = 1, 2, \dots, N$ . The actual return on the portfolio of multiple assets over some specific time period is straightforwardly calculated as follows:

$$R_P = w_1 r_1 + w_2 r_2 + \dots + w_N r_N \quad (9)$$

The expected portfolio return is the weighted average of the expected return of each asset in the portfolio. The weight assigned to the expected return of each asset is the percentage of the market value of the asset to the total market value of the portfolio. Therefore, the expected return  $E(R_P) = \mu_P$  of the portfolio at the end of the period  $t$  is calculated as follows:

$$E(R_P) = w_1 E(r_1) + w_2 E(r_2) + \dots + w_N E(r_N) = \sum_{i=1}^N w_i \mu_i \quad (10)$$

Variance of return for the portfolio used above part as follows:

$$\text{Var}(R_P) = E(R_P - \mu_P)^2 = E(R_P^2) - \mu_P^2 \quad (11)$$

The variance of the return can be computed from the variance of  $S_t$ ,

$$\text{Var}(R_P) = \text{Var}\left(\frac{S_t - S_0}{S_0}\right) = \frac{1}{S_0^2} \text{Var}(S_t - S_0) = \frac{1}{S_0^2} \text{Var}(S_t) \quad (12)$$

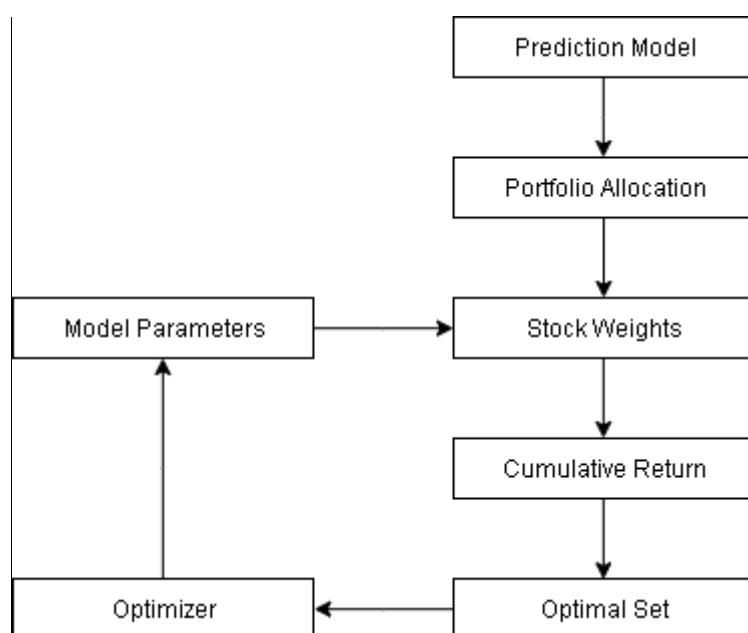
The standard deviations of various random returns is  $\sigma_P = \sqrt{\text{Var}(R_P)}$ . The covariance between asset returns will be denoted by  $\sigma_{ij} = \text{Cov}(r_i, r_j)$ , in particular  $\sigma_{ii} = \sigma_i^2 = \text{Var}(r_i)$ . These are the entries of the  $N \times N$  covariance matrix  $\text{Cov}$ .

$$\text{Cov}(r_i, r_j) = E[(r_i - \mu_i)(r_j - \mu_j)] \quad (13)$$

$$\text{Cov} = \begin{bmatrix} \sigma_{11} & \sigma_{12} & \cdots & \sigma_{1N} \\ \sigma_{21} & \sigma_{22} & \cdots & \sigma_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_{N1} & \sigma_{N2} & \cdots & \sigma_{NN} \end{bmatrix} \quad (14)$$

### 3.2.2. Portfolio Optimization

The objective of portfolio optimization is to try to find the optimal asset allocation based on the stock price prediction phrase. Portfolio construction top-down investing was adapted to pick up the top-performing stock based on the prediction model to construct a multiple asset portfolio, as shown in Figure 4. On the basis of stock prediction results, the expected return and standard deviation for each stock are calculated. For each time period, the top predicted performance stocks with the highest predicted expected returns will be selected to construct a portfolio with initial weights, in which EQ is most commonly assigned. As the number of stocks and the correlated weights are determined, the portfolio cumulative return is calculated by (9). The optimal set is a set of current allocation weights for the selected stocks in the constructed portfolio. By adjusting the model parameters of the portfolio optimizers, we can figure out the optimal weights for the selected stocks in the constructed portfolio. Simulation and optimization techniques were used to seek the optimal weights for the constructed portfolio, instead of using the conventional EQ method.



**Figure 4.** Portfolio optimization model.

#### 1. Simulation Modeling: Monte Carlo Simulation (MCS)

Simulation is a widely used technique for portfolio risk assessment and optimization. Portfolio exposure to different factors is often evaluated over multiple scenarios, and portfolio risk measures such as value-at-risk are estimated. Generating meaningful scenarios is an art as much as science, and presents a number of modeling and computational challenges. Monte Carlo simulation (MCS) is a valuable tool for evaluating functional relationships between variables, visualizing the effect of multiple correlated variables, and testing strategies. MCS solves a deterministic problem based on probabilistic analog by creating scenarios for output variables of interest. First, it generates random portfolio weights and calculates the corresponding portfolio measurements such as expected returns, volatility, and Sharpe ratio. The random weights are adjusted until reaching the highest Sharpe ratio value. All possible generated portfolio scenarios can be seen as a color map as the distribution of random weights in Figure 5a, where an efficient portfolio was found as a red dot sign with the highest Sharpe ratio value. The efficient frontier line is also presented in Figure 5b.

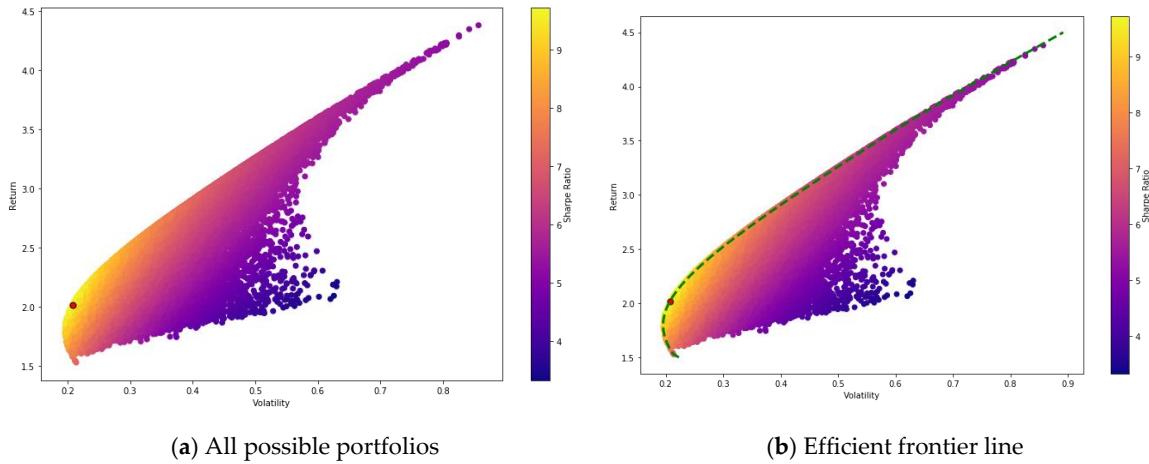


Figure 5. Portfolio efficient frontier.

**Algorithm 1:** Pseudocode of the Monte Carlo Simulation**Input**

- Number of iteration:  $n$
- Number of assets for each portfolio:  $N$
- Initial weight array:  $W_0 = [w_1, \dots, w_N]$  with  $\sum_{i=1}^N w_i = 1$  and  $i = 1, \dots, N$

**Output**

- Maximum of Sharpe ratio
- Optimal weight array  $W = [w_1, \dots, w_N]$  with  $\sum_{i=1}^N w_i = 1$

```
for  $i$  in  $range(n)$ :
```

- Initial random weights:  $W_i$
- Save the temporary weights:  $W_i$
- Calculate expected portfolio return  $exp\_ret[i]$
- Calculate expected volatility  $exp\_vol[i]$
- Calculate Sharpe ratio:  $SR[i] = exp\_ret[i]/exp\_vol[i]$

```
end
```

## 2. Optimization Modeling: Mean-variance Optimization (MVO)

A portfolio constructed from  $N$  different assets can be described by means of the vector of weights  $w = (w_1, w_2, \dots, w_N)$ , with the constraint given  $\sum_{i=1}^N w_i = 1$ . The  $N$ -dimensional vector  $I = (1, 1, \dots, 1)$  is denoted by  $I$ . Therefore, the constraint can conveniently be written as  $w^T I = 1$ . Denote the random returns on the stocks by  $r_1, \dots, r_N$ , and the vector of expected return by  $\mu = (\mu_1, \mu_2, \dots, \mu_N)$  with  $\mu_i = E(r_i)$  for  $i = 1, 2, \dots, N$ . The covariances between returns will be denoted by  $\sigma_{ij} = \text{Cov}(r_i, r_j)$ , in particular  $\sigma_{ii} = \sigma_i^2 = \text{Var}(r_i)$ . These are the entries of the  $N \times N$  covariance matrix  $\Phi$ .

$$\Phi = \begin{bmatrix} \sigma_{11} & \sigma_{12} & \cdots & \sigma_{1N} \\ \sigma_{21} & \sigma_{22} & \cdots & \sigma_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_{N1} & \sigma_{N2} & \cdots & \sigma_{NN} \end{bmatrix} \quad (15)$$

The expected return  $\mu_P = E(R_P)$  and variance  $\sigma_P^2 = \text{Var}(R_P)$  of a portfolio with weights  $w$  are given by

$$\mu_P = \sum_{i=1}^N w_i \mu_i = w^T \mu, \quad (16)$$

$$\sigma_P^2 = \text{Var}(R_P) = \sum_{i,j=1}^N w_i w_j \sigma_{ij} = w^T \Phi w. \quad (17)$$

The classical mean-variance portfolio allocation problem is formulated as follows:

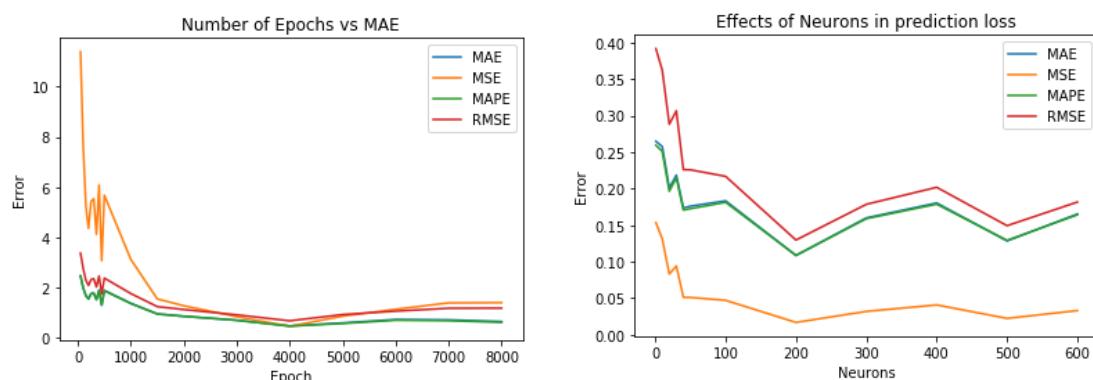
$$\begin{aligned} & \text{Minimize } (w) && w^T \Phi w, \\ & \text{s.t.} && w^T \mu = r_{\text{target}}, \\ & && w^T I = 1. \end{aligned}$$

## 4. Experiment and Results

### 4.1. Data Collection and Experiment Design

In this section, 10-year daily historical stock prices of 500 large-cap stocks listed on the America Stock Exchange Standard & Poor's 500 (S&P 500), which covers nearly 80 percent of the American equity capitalization, was collected by Quandl API from 1 January 2008 till 1 January 2018, with 2516 total trading days. For each stock, daily *Open-High-Low-Close* and *trading volume* was used as the main input values of the dataset. The experiments were conducted on the Ubuntu OS machine containing Intel Core i7-7700 (3.60 GHz) CPU with 64 GB RAM and GeForce GTX 1080 Ti 11176 MB GPU. For model configuration, we used Python 3.6 and Keras library with TensorFlow backend.

In order to set up the hyperparameters for the LSTM prediction model, we first randomly selected one stock from the S&P500 dataset and then performed different measurements. There are two important hyperparameters that might have a high impact on neural network performance, including the number of hidden layers and the number of neurons. We iteratively tuned the number of hidden layers from 2 to 10, and the number of neurons from 1 to 600, to select the optimal model parameters. Prediction loss values were calculated using mean square error (MSE) by adjusting the values of epochs between 5 and 8000; the least loss error was obtained at 4000 epochs. As evident from Figure 6, the minimum prediction loss error was found at 256 and 512 neurons, respectively. A stacked LSTM architecture comprised of 2 LSTM hidden layers was used. As reported in the work of [33], Adam optimization is more suitable for deep learning problems with larger datasets. Therefore, Adam optimizer with default parameters provided by Keras was employed in our experiments. The train–test split ratio used for the LSTM prediction model is 80:20. The detail selected hyperparameters are summarized in Table 1. For machine learning models, the scikit-learn library was used for training prediction models.



**Figure 6.** Model parameters selection. MAE-mean absolute error, MSE-mean square error, MAPE-mean absolute percentage error, RMSE-root mean square error.

**Table 1.** Experiment hyperparameters setup.

| Categories                  | Hyperparameters |
|-----------------------------|-----------------|
| Optimizer                   | Adam            |
| The number of hidden layers | 2               |
| The number of neurons       | 512 and 256     |
| Number of epochs            | 4000            |

#### 4.2. Performance Evaluation

##### 4.2.1. Stock Prediction Evaluation

To evaluate the prediction error rates and model performance, the mean absolute error (MAE) and mean squared error (MSE) were used to measure the difference between the predicted and practical data. MAE and MSE was calculated as follows:

$$MAE = \frac{\sum_{i=1}^T |y_i - y'_i|}{T}, \quad (18)$$

$$MSE = \frac{\sum_{i=1}^T (y_i - y'_i)^2}{T}, \quad (19)$$

where  $Y = (y_1, y_2, \dots, y_T)$  is a vector of actual observations,  $Y' = (y'_1, y'_2, \dots, y'_T)$  is a vector of predicted values, and  $T$  is the number of prediction time horizons.

At first, we conducted two variant types of RNN LSTM and GRU models to predict the stock price. Machine learning models such as LR and SVR were also employed to compare the effectiveness of the LSTM prediction model.

##### 4.2.2. Portfolio Performance Evaluation

On the basis of the prediction results for each time horizon from 1% to 10% of total trading days, as shown in Table 2, a portfolio was constructed  $P_i$  ( $i = 1, 2, \dots, 10$ ) by selecting the top four stocks with the highest predicted returns. For the purpose of optimizing those constructed portfolio performances, simulation modeling and optimization modeling were adopted to select ten efficient portfolios, as represented by  $P_{1,\dots,10}$ , at the final stage by allocating optimal weights. A statistical model for each portfolio was conducted to calculate key factors such as daily return, cumulative return, average daily return, and standard daily return. The cumulative return was used as an identical investment reward to evaluate the performance of each portfolio. However, a high return may come with high volatility or risk in investment. Sharpe ratio (SR) was used for calculating risk-adjusted return, which has used as the industry-standard measurement. Furthermore, active returns of optimal portfolios represented as the difference between the portfolio's actual return on a benchmark were calculated. In our work, the S&P 500 market index was selected as the benchmark. In order to optimize asset allocation for the constructed portfolios, we carried out three different techniques for portfolio optimization:

- Equal-weighted portfolio (EQ) is a type of weighting that gives the same weight to each stock in a portfolio. In our work, we chose initial weight  $w = [.25, .25, .25, .25]$ .
- Monte Carlo simulation (MCS) was used to find the optimal weights of thousands of scenarios or iterations. The number of iterations is  $n = 50,000$ .
- Mean-variance optimization (MVO) was used to find an adaptive weights portfolio that adapted the stock weights using the prediction models.

**Table 2.** Prediction time horizon.

| Period | Start      | End        | Days |
|--------|------------|------------|------|
| 1      | 2017/11/22 | 2017/12/29 | 26   |
| 2      | 2017/10/17 | 2017/12/29 | 51   |
| 3      | 2017/09/12 | 2017/12/29 | 76   |
| 4      | 2017/08/07 | 2017/12/29 | 100  |
| 5      | 2017/06/30 | 2017/12/29 | 126  |
| 6      | 2017/05/25 | 2017/12/29 | 151  |
| 7      | 2017/04/19 | 2017/12/29 | 177  |
| 8      | 2017/03/14 | 2017/12/29 | 202  |
| 9      | 2017/02/06 | 2017/12/29 | 227  |
| 10     | 2016/12/29 | 2017/12/29 | 252  |

#### 4.3. Experiment Results

##### 4.3.1. Stock Prediction Results

At the beginning, portfolios  $P_i (i = 1, \dots, 10)$  were constructed based on the stock prediction models. We evaluated LSTM and GRU prediction models as both are variations of RNN and able to prevent vanishing gradient problems. Table 3 summarized the detailed portfolio as well as the loss of error values for each stock in the constructed portfolios.

**Table 3.** Summarized prediction loss of error. LSTM, long short-term memory; GRU, gated recurrent unit; MAE, mean absolute error; MSE, mean square error.

| P | LSTM  |         |         | GRU   |         |         |
|---|-------|---------|---------|-------|---------|---------|
|   | Stock | MAE     | MSE     | Stock | MAE     | MSE     |
| 1 | DVA   | 0.05965 | 0.00413 | DVA   | 0.04731 | 0.00258 |
|   | FCX   | 0.01899 | 0.00064 | FCX   | 0.02019 | 0.00067 |
|   | KSS   | 0.01364 | 0.00038 | M     | 0.03484 | 0.00164 |
|   | LB    | 0.03459 | 0.00195 | LB    | 0.02525 | 0.00092 |
| 2 | FL    | 0.04111 | 0.00279 | FL    | 0.02515 | 0.00089 |
|   | NKTR  | 0.01227 | 0.00024 | NKTR  | 0.01232 | 0.00028 |
|   | KSS   | 0.01403 | 0.00035 | KR    | 0.04080 | 0.00197 |
|   | LB    | 0.02482 | 0.00105 | LB    | 0.02604 | 0.00105 |
| 3 | MRO   | 0.04735 | 0.00376 | MRO   | 0.01433 | 0.00035 |
|   | NKTR  | 0.04052 | 0.00236 | NKTR  | 0.01565 | 0.00033 |
|   | NATP  | 0.02651 | 0.00124 | SIVB  | 0.01357 | 0.00034 |
|   | LB    | 0.00850 | 0.00016 | LB    | 0.04476 | 0.00302 |
| 4 | GPS   | 0.00321 | 0.00003 | GPS   | 0.01324 | 0.00028 |
|   | NKTR  | 0.00205 | 0.00001 | NKTR  | 0.01554 | 0.00032 |
|   | MU    | 0.00230 | 0.00001 | MU    | 0.01201 | 0.00025 |
|   | URI   | 0.00360 | 0.00002 | URI   | 0.01473 | 0.01554 |
| 5 | GPS   | 0.01071 | 0.00019 | GPS   | 0.01130 | 0.00021 |
|   | NKTR  | 0.01347 | 0.00028 | NKTR  | 0.01567 | 0.00035 |
|   | NRG   | 0.01485 | 0.00048 | NRG   | 0.01461 | 0.00047 |
|   | URI   | 0.01726 | 0.00047 | URI   | 0.01658 | 0.00046 |
| 6 | ALGN  | 0.01432 | 0.00043 | ALGN  | 0.01034 | 0.00026 |
|   | NKTR  | 0.00267 | 0.00002 | NKTR  | 0.00114 | 0.00000 |
|   | NRG   | 0.01930 | 0.00044 | NRG   | 0.00597 | 0.00005 |
|   | TROW  | 0.01360 | 0.00027 | URI   | 0.01389 | 0.00029 |
| 7 | ALGN  | 0.00514 | 0.00004 | ALGN  | 0.00892 | 0.00019 |
|   | NKTR  | 0.00426 | 0.00004 | NKTR  | 0.00367 | 0.00003 |
|   | IPGP  | 0.00294 | 0.00002 | NDVA  | 0.01476 | 0.00037 |
|   | NDVA  | 0.00261 | 0.00002 | TTWO  | 0.01806 | 0.00053 |

**Table 3.** Cont.

| P  | LSTM        |         |         | GRU         |         |         |
|----|-------------|---------|---------|-------------|---------|---------|
|    | Stock       | MAE     | MSE     | Stock       | MAE     | MSE     |
| 8  | <b>ABMD</b> | 0.00813 | 0.00014 | <b>ALGN</b> | 0.01011 | 0.00021 |
|    | <b>ADBE</b> | 0.00756 | 0.00010 | <b>IPGP</b> | 0.00662 | 0.00008 |
|    | <b>ALGN</b> | 0.00161 | 0.00000 | <b>NKTR</b> | 0.00186 | 0.00001 |
|    | <b>AMZN</b> | 0.01258 | 0.00027 | <b>NDVA</b> | 0.01095 | 0.00021 |
| 9  | <b>ALGN</b> | 0.00522 | 0.00006 | <b>ABMD</b> | 0.00847 | 0.00016 |
|    | <b>IPGP</b> | 0.01104 | 0.00030 | <b>ADBE</b> | 0.01157 | 0.00025 |
|    | <b>NKTR</b> | 0.01319 | 0.00027 | <b>ALGN</b> | 0.00919 | 0.00012 |
|    | <b>TTWO</b> | 0.02046 | 0.00063 | <b>ATVI</b> | 0.01863 | 0.00055 |
| 10 | <b>ALGN</b> | 0.00751 | 0.00013 | <b>ALGN</b> | 0.00672 | 0.00010 |
|    | <b>IPGP</b> | 0.01134 | 0.00014 | <b>IPGP</b> | 0.00469 | 0.00005 |
|    | <b>NKTR</b> | 0.00273 | 0.00002 | <b>NKTR</b> | 0.00505 | 0.00004 |
|    | <b>TTWO</b> | 0.00969 | 0.00015 | <b>TTWO</b> | 0.01234 | 0.00027 |

In general, both the LSTM and GRU models had low error values. The LSTM model was found more efficient than the GRU model by obtaining lower error rates. LSTM controls the exposure of memory content (cell state), while GRU exposes the entire cell state to other units in the network. The LSTM unit has separate input and forget gates, while the GRU performs both of these operations together via its reset gate.

#### 4.3.2. Portfolio Performance Evaluation

Secondly, we evaluated the constructed portfolios based on prediction models, including SVR, LR, and LSTM models. In order to optimize the performance of the constructed portfolio, MCS and MVO were employed to evaluate the impact of optimization on portfolio performances. In the majority of cases, expected returns showed a tendency to increase, while SRs tended to decline gradually over time, except in the SVR model, where they fluctuated significantly. It can be observed in Figures 7–9 that MCS and MVO techniques perform approximately the same in all built prediction models. As predicted by the SVR prediction model in Figure 7, expected returns fluctuated. Although SRs obtained by MCS and MVO methods were considerably higher than EQ, the performance of constructed portfolios showed poorer performance compared with the LR and LSTM models in both expected returns and SRs. For example, the highest expected return and SR were only 55% and 0.05 at  $P_8$  and  $P_{10}$ , respectively. As shown in Figure 8, portfolios constructed by the LR model obtained the highest expected returns using the EQ method compared with MCS and MVO over periods. However, the constructed portfolios obtained higher SRs using MCS and MVO compared with the EQ method. This suggests that optimization methods can improve the performance of constructed portfolios by increasing the SR values; in other words, optimization techniques are not inevitably guaranteed to improve the return but can reduce the risk in trading. As reported in Figure 9, our constructed portfolios based on the LSTM prediction model obtained the highest expected returns as well as SRs in most of the predicted periods. EQ showed the effectiveness of returns, however, the gap between MCS and MVO was smaller than EQ, as shown in Figure 9. Therefore, the LSTM prediction model is more efficient than the SVR and LR models.

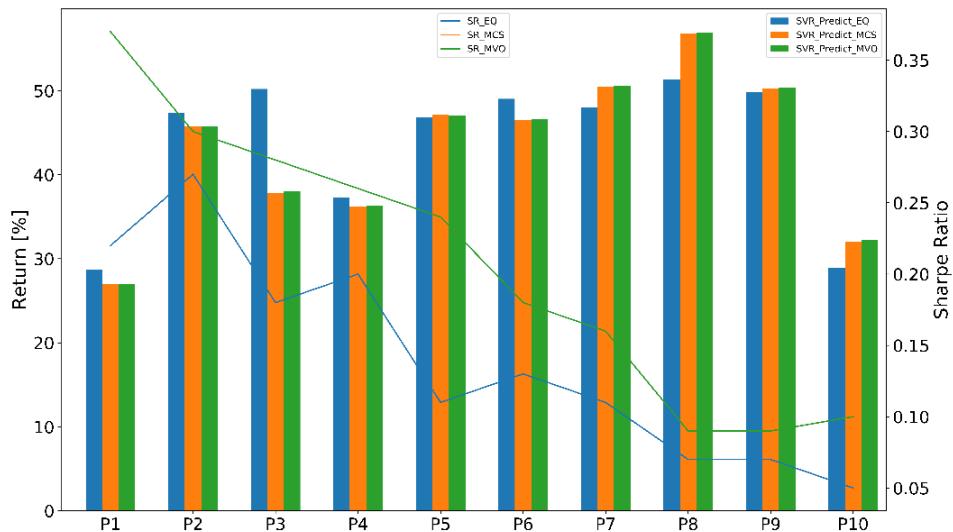


Figure 7. Support vector regression (SVR) constructed portfolios' performance.

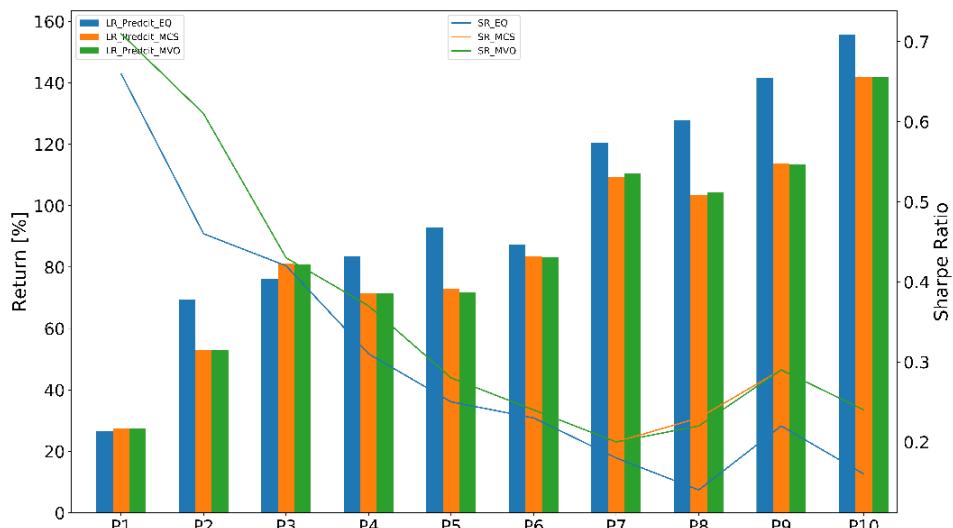


Figure 8. Linear regression (LR) constructed portfolios' performance.

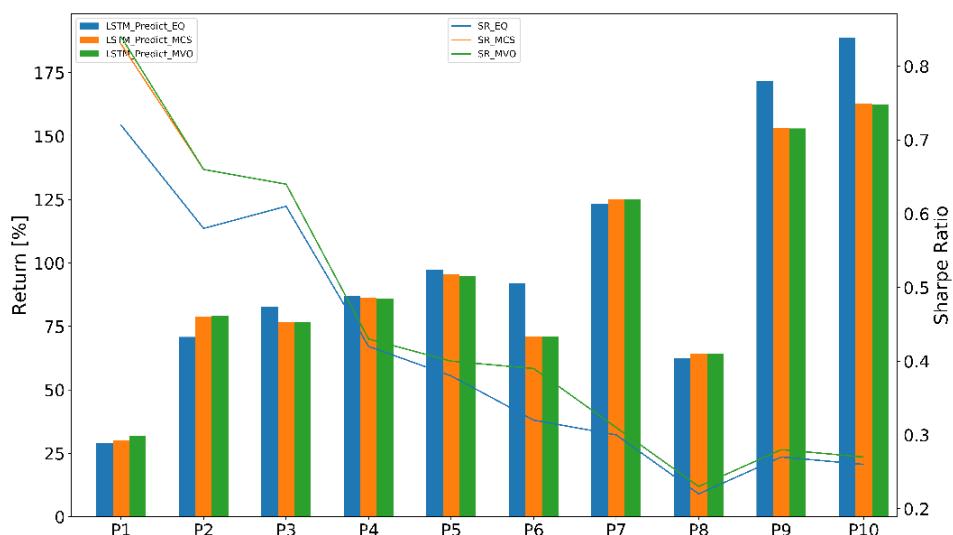


Figure 9. LSTM constructed portfolios' performance.

Third, the constructed portfolios and weights for each portfolio in the prediction phase were tested in actual trading. The constructed portfolios' performance based on prediction models is presented in Figures 10–12. Actual returns and SRs based on the SVR prediction model fluctuated significantly on time periods. MCS and MVO weights showed effective improvements in the returns but showed less impact on SRs. Especially, the actual return and SR suddenly drop to negative in  $P_8$ , as shown in Figure 10. The fluctuation can be the cause of low predicted accuracy. Therefore, predicted stocks and adjusted weights were ineffective. Constructed portfolios based on the LR prediction model performed better than those based on the SVR prediction model in practical trading, as shown in Figure 11, and EQ weights obtained higher returns compared with MCS and MVO. Unfortunately, SRs produced by MCS and MVO weights gradually decreased, even lower than EQ weights. The results seem to indicate that the predicted accuracy contributed a considerable impact on the optimization phase. In this phase, constructed portfolios based on the LSTM model outperformed, as shown in Figure 12, with the highest returns and SRs. On one hand, the returns from EQ, MCS, and MVO were pretty much the same in almost cases. On the other hand, SRs obtained by MCS and MVO were higher compared with EQ weights. It is apparent that the constructed portfolio based on the LSTM prediction model outperformed the proposed prediction models. The higher the accuracy obtained, the higher the return and reliable risk control we can construct.

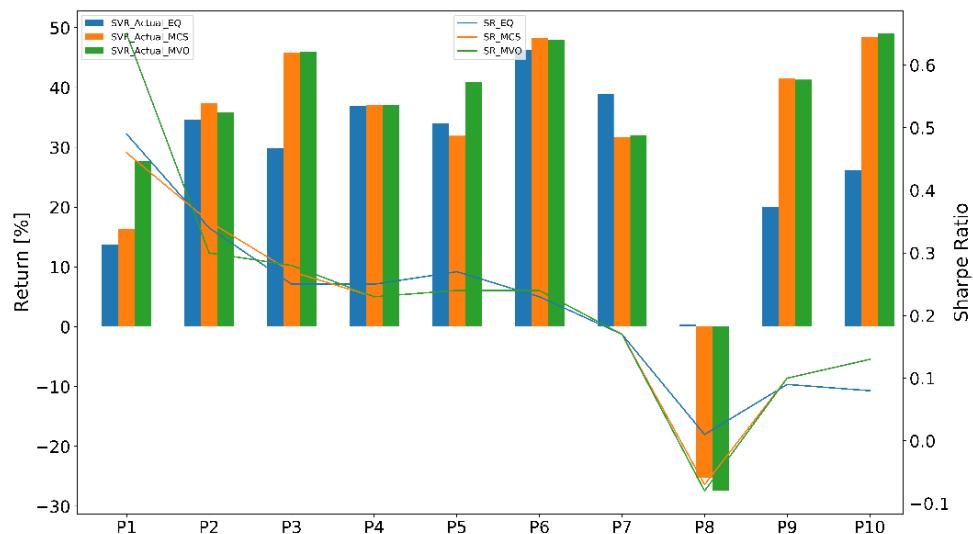


Figure 10. SVR actual portfolios' performance.

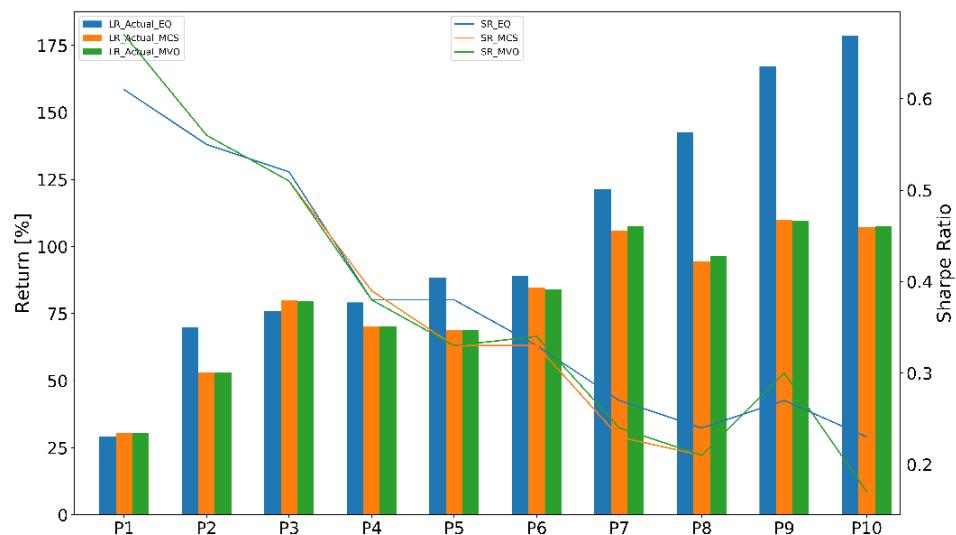


Figure 11. LR actual portfolios' performance.

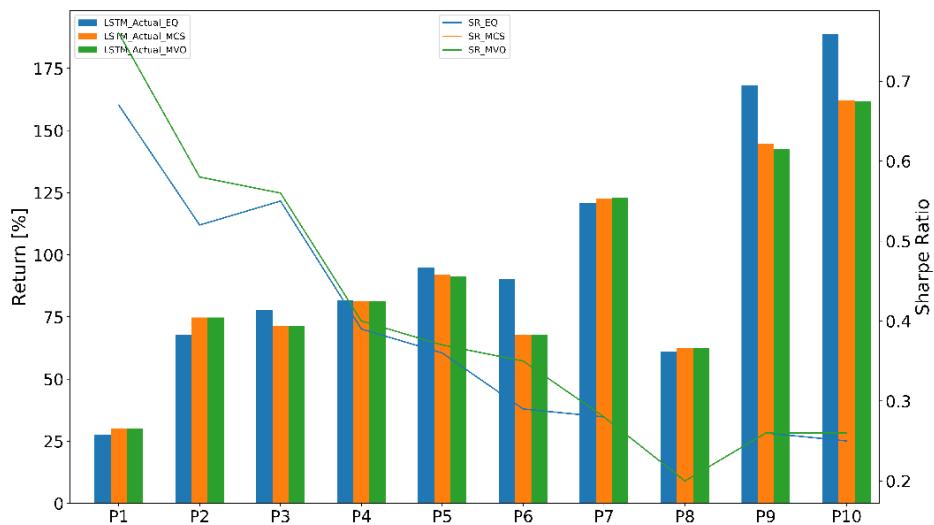


Figure 12. LSTM actual portfolios' performance.

After evaluating the efficiency of the proposed prediction models in both prediction and practical trading. The results showed that the LSTM prediction model outperformed the proposed prediction models. The efficient prediction is not only for prediction but also support for the optimization phase. These constructed portfolios were selected as efficient portfolios for quantitative trading. As we can see from the results, MCS and MVO weights were slightly different, however, as the number of iterations (scenarios) increases, more computational resources are required. Therefore, MVO weights were selected as the optimal weights for efficient portfolios. The comparison of efficient portfolio performance is given in Figure 13. Returns were gradually increased in both prediction and practical trading. In addition, returns obtained in practical trading are correlated with predicted results. Although SRs showed a tendency to decline, there was only a slight difference between prediction and practical trading. This is evidence that efficient portfolios beat the benchmark S&P 500 in both returns and SRs. Figure 14 shows a summary of the optimal portfolio allocation based on the LSTM prediction model. It showed that the optimal weights of stocks in the portfolio lead to higher active returns and lower volatility relative to the benchmark index. All these differences were statistically significant at approximately 86 and 48 percent higher than the benchmark in terms of return and SR, respectively. Table 4 summarizes the comparison between the efficient portfolios and the benchmark on active returns for each time period. Our constructed portfolios outperformed the benchmark the S&P 500 index.

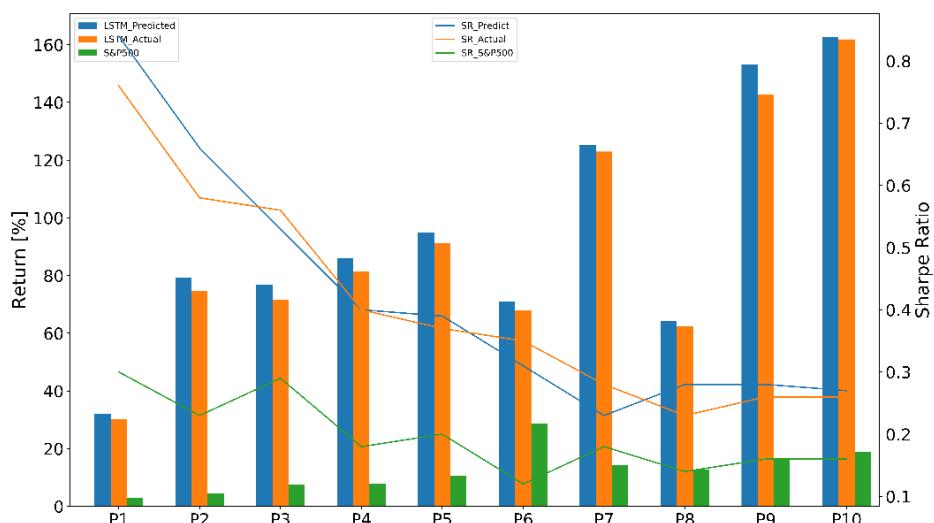
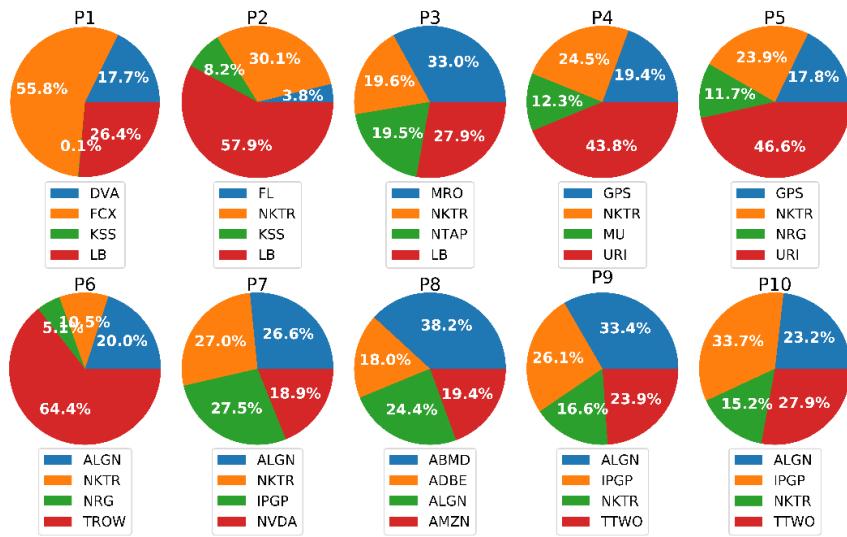


Figure 13. LSTM efficient portfolios' performances versus the benchmark index.



**Figure 14.** The detailed LSTM optimal allocation for each efficient portfolio.

**Table 4.** Optimal portfolio returns.

| P  | Portfolio [%] | Benchmark [%] | Active Return [%] |
|----|---------------|---------------|-------------------|
| 1  | 30.09         | 2.95          | 24.71             |
| 2  | 74.77         | 4.53          | 70.23             |
| 3  | 71.51         | 7.46          | 64.06             |
| 4  | 81.36         | 7.77          | 73.59             |
| 5  | 91.31         | 10.49         | 80.82             |
| 6  | 67.89         | 28.78         | 39.11             |
| 7  | 122.94        | 14.15         | 108.79            |
| 8  | 62.40         | 12.65         | 49.75             |
| 9  | 142.59        | 16.73         | 126.22            |
| 10 | 161.84        | 18.83         | 143.01            |

## 5. Conclusions and Discussions

Stock prediction plays a significant role in constructing an investment portfolio in terms of two important aspects stock selection and allocation. This paper presented the LSTM network, a type of recurrent neural network, to predict the stock price in order to demonstrate a typical quantitative trading strategy. The proposed model works efficiently by achieving high accuracy compared with other machine learning such as LR and SVM. As a result, we can take advantage of the prediction results to construct a quantitative portfolio for each predicted time horizon. On the basis of the optimization techniques, our constructed portfolios performed effectively by obtaining high returns in both prediction and actual trading, as well as compared with the S&P 500 index. According to the performance of constructed portfolios, the active returns are in inverse proportion to Sharpe ratio values, which can be understood as a fact of risk-return trade-offs in existence in our work. A prediction model that combines a strategic prediction based on historical data with a dynamic prediction, based on valuation, momentum, and spillover, should be extensively investigated in order to minimize the risk-return trade-offs. Furthermore, dynamic portfolio optimization and diversification are also considered as the target for further research that allows designing multiple tactical, flexible trading strategies in order to maximize trading profits.

There are several challenges for building effective quantitative trading strategies through deep learning. First, market data exhibit a high noise to signal ratio. The prediction models can perform well on the historical data set. However, the stock market always fluctuates as a result of factors such as market psychology, macroeconomics, and even political issues. Therefore, high performance on the historical dataset does not guarantee earning a desirable profit in practical trading. Second,

backtesting is not only a tool to evaluate the discovered strategy but also helps to avoid false positives. Finally, developing a flexible, efficient trading strategy is critically important for quantitative trading. It is one of the most challenging tasks in the quantitative trading system. Diverse data sources and formats, as well as different characteristics of data, are causing the prediction task to become more complex. In summary, the deep learning approach shows a remarkable effect on stock prediction performance, which can be an essential condition for portfolio construction and optimization process in quantitative trading.

**Author Contributions:** Conceptualization, V.-D.T. and C.-M.L.; methodology, V.-D.T. and D.A.T.; software, V.-D.T. and D.A.T.; validation, V.-D.T.; formal analysis, V.-D.T. and C.-M.L.; investigation, V.-D.T.; resources, C.-M.L.; data curation, V.-D.T.; writing—original draft preparation, V.-D.T. and D.A.T.; writing—review and editing, V.-D.T. and C.-M.L.; visualization, V.-D.T. and D.A.T.; supervision, C.-M.L.; project administration, C.-M.L.; funding acquisition, C.-M.L. All authors have read and agreed to the published version of the manuscript.

**Funding:** This work is partially supported by the National Taipei University of Technology under the grant: NTUT-BIT-108-02.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Fabozzi, F.J.; Markowitz, H.M. *The Theory and Practice of Investment Management: Asset Allocation, Valuation, Portfolio Construction, and Strategies*, 2nd ed.; John Wiley and Sons: Hoboken, NJ, USA, 2011; Volume 198, pp. 289–290.
2. Adebiyi, A.A.; Adewumi, A.O.; Ayo, C.K. Comparison of ARIMA and artificial neural networks models for stock price prediction. *J. Appl. Math.* **2014**. [[CrossRef](#)]
3. Cumming, J.; Alrajeh, D.D.; Dickens, L. An Investigation into the Use of Reinforcement Learning Techniques Within the Algorithmic Trading Domain. Master’s Thesis, Imperial College London, London, UK, 2015.
4. Chong, E.; Han, C.; Park, F.C. Deep learning networks for stock market analysis and prediction: Methodology, data representations, and case studies. *Expert Syst. Appl.* **2017**, *83*, 187–205. [[CrossRef](#)]
5. Fabozzi, F.J.; Pachamanova, D.A. *Portfolio Construction, and Analytics*; John Wiley & Sons: Hoboken, NJ, USA, 2016; pp. 111–112.
6. Kissell, R.L. *The Science of Algorithmic Trading and Portfolio Management*; Academic Press: Cambridge, MA, USA, 2013; pp. 111–112.
7. Ta, V.D.; Liu, C.M.; Addis, D. Prediction and Portfolio Optimization in Quantitative Trading Using Machine Learning Techniques. In Proceedings of the Ninth International Symposium on Information and Communication Technology, Da Nang, Vietnam, 6–7 December 2018; pp. 98–105.
8. Six Asset Allocation Strategies that Work. Available online: <https://www.investopedia.com/investing/6-asset-allocation-strategies-work> (accessed on 4 October 2019).
9. Markowitz, H. Portfolio selection. *J. Financ.* **1952**, *7*, 779–781.
10. Sharpe, W.F.; Sharpe, W.F. *Portfolio Theory and Capital Markets*; McGraw-Hill: New York, NY, USA, 1970; Volume 217.
11. Roll, R.; Ross, S.A. An empirical investigation of the arbitrage pricing theory. *J. Financ.* **1980**, *35*, 1073–1103. [[CrossRef](#)]
12. Fama, E.F.; French, K.R. Common risk factors in the returns on stocks and bonds. *J. Financ. Econ.* **1993**, *33*, 35–36. [[CrossRef](#)]
13. He, G.; Litterman, R. *The Intuition Behind Black-Litterman Model Portfolios*; Goldman Sachs Investment Management Research: New York, NY, USA, 1999.
14. Kolm, P.N.; Tutuncu, R.; Fabozzi, F.J. 60 Years of portfolio optimization: Practical challenges and current trends. *Eur. J. Oper. Res.* **2014**, *234*, 356–371. [[CrossRef](#)]
15. Ahmadi-Javid, A.; Fallah-Tafti, M. Portfolio optimization with entropic value-at-risk. *Eur. J. Oper. Res.* **2019**, *279*, 225–241. [[CrossRef](#)]
16. Lejeune, M.A.; Shen, S. Multi-objective probabilistically constrained programs with variable risk: Models for multi-portfolio financial optimization. *Eur. J. Oper. Res.* **2016**, *252*, 522–539. [[CrossRef](#)]
17. Lwin, K.T.; Qu, R.; MacCarthy, B.L. Mean-VaR portfolio optimization: A nonparametric approach. *Eur. J. Oper. Res.* **2017**, *260*, 751–766. [[CrossRef](#)]

18. Qin, Z. Mean-variance model for portfolio optimization problem in the simultaneous presence of random and uncertain returns. *Eur. J. Oper. Res.* **2015**, *245*, 480–488. [[CrossRef](#)]
19. Samarakoon, L.P.; Hasan, T. *Portfolio performance evaluation*. *Encyclopedia of Finance*, 2nd ed.; Springer: New York, NY, USA, 2006; pp. 617–622.
20. Aragon, G.O.; Ferson, W.E. Portfolio performance evaluation. *Found. Trends Financ.* **2007**, *2*, 831–890. [[CrossRef](#)]
21. Elleuch, J.; Trabelsi, L. Fundamental analysis strategy and the prediction of stock returns. *Int. Res. J. Financ. Econ.* **2009**, *30*, 95–107.
22. Sezer, O.B.; Ozbayoglu, A.M.; Dogdu, E. An artificial neural network-based stock trading system using technical analysis and big data framework. In Proceedings of the South East Conference, Haines, AK, USA, 4–12 April 2017; pp. 223–226.
23. Fang, L.; Yu, H.; Huang, Y. The role of investor sentiment in the long-term correlation between US stock and bond markets. *Int. Rev. Econ. Financ.* **2018**, *58*, 127–139. [[CrossRef](#)]
24. Nguyen, T.H.; Shirai, K.; Velcin, J. Sentiment analysis on social media for stock movement prediction. *Expert Syst. Appl.* **2015**, *42*, 9603–9611. [[CrossRef](#)]
25. Lam, M. Neural network techniques for financial performance prediction: Integrating fundamental and technical analysis. *Decis. Support Syst.* **2004**, *37*, 567–581. [[CrossRef](#)]
26. Deng, S.; Mitsubuchi, T.; Shioda, K.; Shimada, T.; Sakurai, A. Combining technical analysis with sentiment analysis for stock price prediction. In Proceedings of the 2011 IEEE Ninth International Conference on Dependable, Autonomic and Secure Computing, Sydney, Australia, 12–14 December 2011; pp. 800–807.
27. Sirignano, J.A. Deep learning for limit order books. *Quant. Financ.* **2019**, *19*, 549–570. [[CrossRef](#)]
28. Sohangir, S.; Wang, D.; Pomeranets, A.; Khoshgoftaar, T.M. Big Data: Deep Learning for financial sentiment analysis. *J. Big Data* **2018**, *5*, 3. [[CrossRef](#)]
29. Xiong, R.; Nichols, E.P.; Shen, Y. Deep Learning Stock Volatility with Google Domestic Trends. *arXiv* **2015**, arXiv:1512.04916.
30. Heaton, J.B.; Polson, N.G.; Witte, J.H. Deep learning for finance: Deep portfolios. *Appl. Stoch. Models Bus. Ind.* **2017**, *33*, 3–12. [[CrossRef](#)]
31. Bao, W.; Yue, J.; Rao, Y. A deep learning framework for financial time series using stacked autoencoders and long-short term memory. *PLoS ONE* **2017**, *12*, e0180944. [[CrossRef](#)]
32. Shen, F.; Chao, J.; Zhao, J. Forecasting exchange rate using deep belief networks and conjugate gradient method. *Neurocomputing* **2015**, *167*, 243–253. [[CrossRef](#)]
33. Chung, J.; Gulcehre, C.; Cho, K.; Bengio, Y. Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling. *arXiv* **2014**, arXiv:1412.3555.
34. Fischer, T.; Krauss, C. Deep learning with long short-term memory networks for financial market predictions. *Eur. J. Oper. Res.* **2018**, *270*, 654–669. [[CrossRef](#)]
35. Nguyen, T.T.; Yoon, S. A Novel Approach to Short-Term Stock Price Movement Prediction using Transfer Learning. *Appl. Sci.* **2019**, *9*, 4745. [[CrossRef](#)]



© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).