A Modified LSTM Neural Network Based on Boosting and Sufficient Dimensionality Reduction

Zhikun Chen [1, #], Shixuan Lin [2, #], Xinrui Kou [3, *, #]

[1] College of Economics, Shenzhen University, Shenzhen, China, 518055

[2] School of Economics and Management, Beihang University, Beijing, China, 100191

[3] College of Economics, Hebei University, Baoding, China, 071000

* Corresponding Author Email: k1020816131@163.com

[#]These authors contributed equally

Abstract. Stock price prediction is a common scenario in time series data forecasting, providing effective guidance for investment decisions. Long Short-Term Memory (LSTM) is a widely used model for stock price prediction, yet the selection of its hyperparameters remains an unresolved issue. In this paper, we address this challenge by employing model averaging instead of model selection. Specifically, we adaptively solve the hyperparameter selection problem by utilizing a distance covariance-weighted method, effectively balancing the bias and variance of the predictive model. Additionally, we propose an enhanced model that employs a boosting approach based on sufficiently reducing dimensionality through a multifactor model. This approach captures stock price sequence information beyond volatility. Practical data analysis demonstrates that the proposed method exhibits significant advantages over the original LSTM model in terms of mean square error or absolute error. Furthermore, the proposed framework can be applied to hyperparameter selection in other time series prediction models, such as autoregressive integrated moving averages (ARIMA), including the selection of autoregressive and partial autocorrelation orders.

Keywords: boosting, sufficient dimensionality reduction, stock price prediction, LSTM, model averaging.

1. Introduction

In recent years, with the flourishing development of the financial industry in China, stock price prediction has become a closely discussed topic among industry scholars. For investors, higher accuracy in stock price prediction corresponds to lower investment risks. Therefore, establishing predictive models for stock data has been a hot topic in the field of quantitative finance.

It is well-known that stock price time series typically exhibit nonlinear and complex volatility trends. With the rapid development of computer technology and deep learning, neural network-based stock price prediction models have demonstrated more accurate results compared to conventional multifactor prediction models. For instance, in 1988, the earliest application of the BP neural network for stock research predicted the daily stock returns of IBM [1]. The use of Convolutional Neural Networks (CNN) demonstrated good prediction results by recognizing patterns and extracting features for predicting stock trends [2]. Neural network models do not require cumbersome assumptions and model structures; they can establish models as long as sufficient data is provided. Through processing in the hidden layers, their ability to handle nonlinear data is enhanced. Additionally, Recurrent Neural Networks (RNN) address the task of capturing temporal dependencies through neuron connections within the same layer, overcoming the limitations of artificial neural networks.

In recent years, many scholars have innovatively combined various individual models, further improving prediction performance. For example, in 2014, Kristjanpoller et al. integrated Artificial Neural Networks (ANN) and the Generalized Autoregressive Conditional Heteroskedasticity (GARCH) model to propose a volatility prediction model for the Latin American market [3]. The results indicated that this model had a smaller mean square error compared to the GARCH model.

Patel, Shah, Thakkar, and Kotecha established a hybrid model of ANN, Random Forest, and Support Vector Regression (SVR) to predict the Indian stock market [4]. However, RNN faces optimization issues; if the hidden layers are too numerous, it may fail to remember information from a long time ago, leading to gradient vanishing and exploding. Hence, in 1997, Hochreiter and Schmidhuber introduced the Long Short-Term Memory (LSTM) model, addressing these problems by incorporating input gates, output gates, and forget gates to characterize long-term memory features [5]. It selectively remembers and interacts with time series information, making it suitable for addressing the randomness and non-stationarity of stock prices, demonstrating excellent predictive capabilities [6,7].

However, in practical applications, the LSTM model faces two issues. On one hand, there is the problem of hyperparameter selection, which involves choosing an appropriate window width. If the window width is too large, it may fail to capture local details, while if it is too small, the information captured may be overfitting. On the other hand, apart from stock price volatility information, there is still information in the residual sequence that the LSTM model has not extracted. Therefore, in this paper, we first use the model averaging method to replace the model selection approach, addressing the window selection problem and effectively balancing the variance and bias of the predictive model. Secondly, to capture information beyond volatility, we establish a multifactor prediction model based on sufficient dimensionality reduction. This model predicts the residual sequence of LSTM, capturing information from the residual sequence. Practical data analysis results demonstrate that the proposed method significantly enhances predictive performance compared to the initial LSTM model.

2. The basic theory and methodology

2.1. Long Short-Term Memory (LSTM) Neural Network

The LSTM neural network is a special type of recurrent neural network (RNN) that is less susceptible to the issues of gradient vanishing and exploding. In comparison to regular RNNs, LSTM effectively selects key features, can retain early information through the control of "gates," and exhibits better performance in longer sequences, possessing enhanced feature storage capabilities and more accurate historical information retrieval advantages [8].

The structure of a single LSTM cell layer is depicted in Figure 1, primarily comprising forget gates, input gates, and output gates. Here, X_t represents the input data for the current state, h_{t-1}, and h_t are the output values of the hidden layer at the previous time point and the current time point, C_{t-1}, and C_t are the cell states at the previous and current time points, f_t is the output value of the forget gate, i_t and $\tilde{C}_t$ represent the output values of the activation functions σ and tanh, and o_t is the output result of the output gate at the current time.

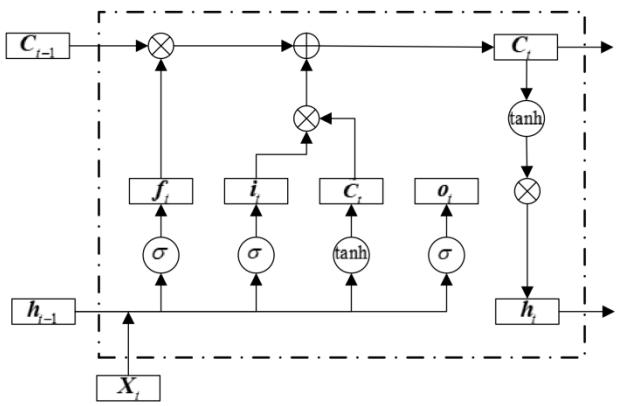

Figure 1: Illustrates the structure of the LSTM layer.

2.2. Model Averaging Method

The model averaging method combines multiple models without readily excluding any model, thus generally reducing the loss of useful information. Additionally, model averaging provides a safeguard mechanism, avoiding the selection of poorly performing models.

The development of model averaging methods mainly follows two directions: Bayesian Model Averaging (BMA) and Frequentist Model Averaging (FMA). The basic steps of BMA are as follows: set prior probabilities for the models to be combined and prior distributions for the parameters within each model, then perform statistical inference using classical Bayesian methods. Challenges in BMA include determining the prior probabilities for each model, as different priors can significantly impact BMA results. Another challenge is the potential introduction of conflicting priors for the parameters to be estimated, leading to difficulties in interpretation and acceptance [9]. Additionally, BMA typically assumes that the true model is among the considered candidate models, but in reality, the true model is often more complex. Considering these issues with BMA, FMA methods have gained increasing attention and research. In the study of FMA methods, the most crucial issue is the selection of combination weights.

In this paper, we use the distance correlation coefficient as weights to average LSTM models with different window widths.

$$\mu = \sum w_i LSTM_i \tag{1}$$

$$w = dCor(X, Y) \tag{2}$$

$LSTM_i$ represents the prediction results with different window width parameters, w_i denotes their respective weights, and μ is the weighted average result of LSTM model predictions for different window widths, where the weights are determined by the distance correlation coefficients.

There are various definitions of distance correlation coefficients, and equivalence exists among different definitions. Specifically, assuming (X_i, Y_i), $i=1\ldots, n$ are n random samples of the random vector (X, Y), the distance correlation coefficient between X_i and Y_i is defined as:

$$dCor(X, Y) = \frac{dCov(X, Y)}{\sqrt{dVar(X)dVar(Y)}} \tag{3}$$

Here, $dCov(X, Y)$ is the distance covariance coefficient between X and Y; $dVar(X)$ and $dVar(Y)$ are the distance variance coefficients for X and Y, respectively, with $dVar(X) = dCov(X, X)$. The sample distance correlation coefficient is expressed as:

$$dCor_n(X, Y) = \frac{dCov_n(X, Y)}{\sqrt{dVar_n(X)dVar_n(Y)}} \tag{4}$$

Here, $dCov_n^2(X, Y) = \frac{1}{n^2}\sum_{j,k=1}^{n} A_{jk} B_{jk}$, $dVar^2(X) = dCov_n^2(X, X) = \frac{1}{n^2}\sum_{j,k=1}^{n} {A_{jk}}^2$, $A_{jk} = a_{jk} - \bar{a}_{j.} - \bar{a}_{.k} + \bar{a}_{..}$, $B_{jk} = b_{jk} - \bar{b}_{j.} - \bar{b}_{.k} + \bar{b}_{..}$, where $a_{jk} = \| X_j - X_k \|$, $b_{jk} = \| Y_j - Y_k \|$ are Euclidean distance metrics, $\bar{a}_{j.}$ is the mean of the j^{th} row, $\bar{a}_{.k}$ is the mean of the k^{th} column, $\bar{a}_{..}$ is the overall mean.

2.3. Sufficient Dimensionality Reduction Theory

Dealing with non-parametric regression models and analysis under high-dimensional variables, as in this paper, poses a significant challenge. To address this issue, sufficient dimensionality reduction proves to be an effective method. Considering the regression function of the dependent variable Y concerning explanatory variables $x_1 \ldots x_p$:

$$Y = g(\beta_1^T x, \cdots, {\beta_K}^T x, \varepsilon) \tag{5}$$

Where $x=(x_1,\ldots x_p)^T$ is the independent variable, ε represents the error term of the regression function, K is the dimensionality of the reduced space, and there is $K << p$. The relationship between

Y and x, as well as β_i, $i=1,2,\ldots K$, are not given. When the dimension of x is large, it becomes difficult for us to directly seek the expression of the relationship between Y and $\beta_1^T x,\ldots, \beta_K^T x$ from $x=(x_1,\ldots x^p)^T$.

However, when we know the forms of $\beta_1,\ldots, \beta_K$, it becomes easy to find the relationship expression between Y and $\beta_1^T x,\ldots, \beta_K^T x$. Therefore, we no longer directly start from $x=(x_1,\ldots x_p)^T$ to find the relationship expression between Y and x, but start from seeking $\beta_1,\ldots, \beta_K$ to find the relationship expression between Y and x. Record $B=(\beta_1,\ldots \beta_K)$, and use $M(A)$ to denote the column space of a matrix A of size $K \times K$. Due to the unknown expression of $g(.)$, the matrix $B = (\beta_1,\ldots \beta_K)$ is not unique. We call the process of seeking $M(B)$ estimation a sufficient dimensionality reduction process, and $\beta_1,\ldots, \beta_K$ are called dimensionality reduction vectors. Based on the above, we can record the relationship between Y and $\beta_1^T x,\ldots, \beta_K^T x$ as follows:

$$\begin{aligned} Y &= g(\beta_1^T x, \cdots, \beta_k^T x, \varepsilon) = g(B^T x, \varepsilon) \\ &= f(x) = g((A^T)^{-1}(BA)^T x, \varepsilon) \triangleq g^*((BA)^T x, \varepsilon) \end{aligned} \tag{6}$$

From the above expression, it can be observed that the column vectors of BA can also be referred to as reduction vectors, indicating that the reduction vectors are not uniquely determined. Therefore, it is sufficient to seek a set of basis vectors for the vector space formed by $\beta_1,\ldots \beta_K$. This, in essence, is the problem of sufficient dimensionality reduction.

2.4. Model Establishment Process

As stock prices exhibit a time series nature, we initially employ the LSTM model to predict this time series. The window width, an essential parameter of the LSTM model, also known as the lag period, determines how many periods of data the LSTM model uses as independent variables. The choice of window width is crucial; if it is too large, the final prediction result may underfit, and if it is too small, the result may overfit. Therefore, we use multiple LSTM models with different window widths for model averaging to address this parameter selection issue.

The use of the LSTM model for stock price prediction aims to capture the inherent volatility of stock prices. However, besides volatility, there are other pieces of information in the stock price sequence that the LSTM model might not capture. These pieces of information remain in the residual sequence obtained from the LSTM model. Hence, we use a multifactor model to fit this residual sequence.

We first explore potential factors that may determine stock price trends. Then, using a sufficient dimensionality reduction method, we reduce factors that may exhibit multicollinearity to mutually uncorrelated factors. Subsequently, we employ Bagging Random Forest to predict the residual sequence, and the obtained residual sequence is used to improve the prediction results of the LSTM model.

3. Data Analysis

3.1. Data Source and Experimental Setup

The experimental data is sourced from Choice Financial Terminal, covering daily data for factors such as volatility, stock price, shareholder-related factors, scale factors, and trading factors for three stocks—China Communications Construction, Kevin Education, and Hua Shu Media—from October 9, 2020, to September 28, 2023 (a total of 728 days).

The experiment is conducted using the Python programming language in a Jupyter Notebook environment. The division between training and testing sets is based on the total sample count (200 samples in this experiment for training and testing), with a boundary point list set to optimize by iteratively searching for the optimal boundary point to ensure the reliability of experimental results. The window width is dynamically adjusted based on the forecast period and sample count to better capture the local features of the data. The weighted average of LSTM models with different window

widths is performed using the reciprocal of mean square error or distance covariance as weights to obtain robust results.

Error calculation involves two methods: mean square error (MSE) and mean relative error (MRE), providing a comprehensive assessment of the model's performance on both the training and testing sets. This experimental setup aims to leverage the data fully, optimize the model training process, and comprehensively evaluate its predictive capabilities.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{7}$$

$$MRE = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{y_i - \hat{y}_i}{y_i}\right| \tag{8}$$

3.2. Comparative Results

In this study, we employed a comparative approach by conducting an in-depth comparison between the original LSTM model and our proposed modified LSTM neural network based on Boosting and sufficient dimensionality reduction. Through a detailed comparison of mean square error (MSE) and absolute error (MRE) under different training sets, we presented a comprehensive view of the superiority of our method in practical applications compared to the original LSTM model. As shown in Table 1 (note that the two vertical axes for MSE and MRE in the chart are independent, with different units), our method achieved significant improvements in both MSE and MRE metrics relative to the original LSTM model. It can be observed that when the training set and test set sample split point are in the interval (90, 95), the improved mean square error and absolute error are both at the bottom. Therefore, setting the training set and test set boundary point at the 45-50 percentile range appears to be optimal.

Table 1: Comparison of Results between LSTM and Modified LSTM

Index	bLSTM-MSE	bLSTM -MRE	LSTM-MSE	LSTM-MRE
75	0.352761	0.061736	1.350759	0.109064
80	0.336711	0.059606	0.783378	0.072377
85	0.321605	0.057579	2.601978	0.162529
90	0.310985	0.05532	1.763844	0.128198
95	0.314955	0.054975	1.04961	0.089719
100	0.347101	0.057369	2.578019	0.159618
105	0.339173	0.055688	1.234744	0.099606
110	0.364967	0.058014	2.101412	0.13892
115	0.382289	0.058441	1.658551	0.120046
120	0.413002	0.060308	2.872728	0.16704
125	0.443329	0.060979	2.697757	0.160792

The significant performance improvement stems from achieving an ideal balance between bias and variance in different models. Our multifactor model not only effectively captures volatility information but also demonstrates outstanding performance in capturing other key information in stock price prediction. In the process of LSTM prediction, we use distance correlation coefficient as weights to average different window widths of LSTM models, resulting in more scientifically grounded predictions. As can be seen from Figures 2 and 3, this improvement not only theoretically provides a more accurate representation of data for the model but is also thoroughly validated in practical results. The substantial improvement in MSE and MRE further confirms the excellence of our approach to stock price prediction.

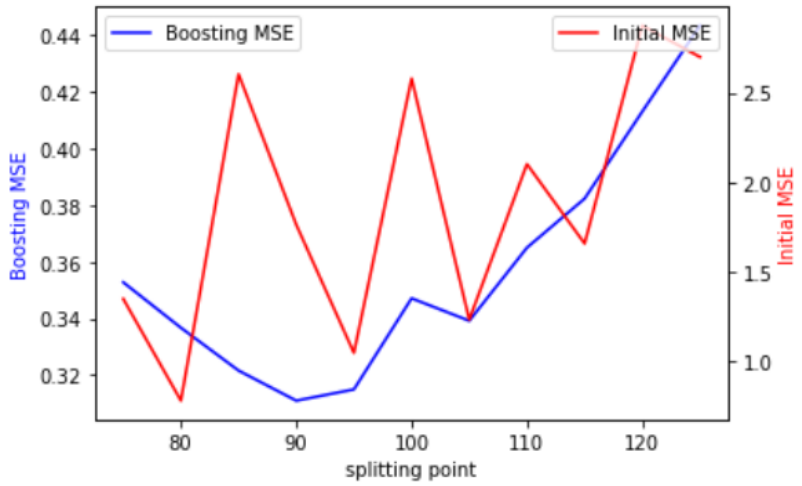

Figure 2: Comparison of Mean Square Error between LSTM and Modified LSTM

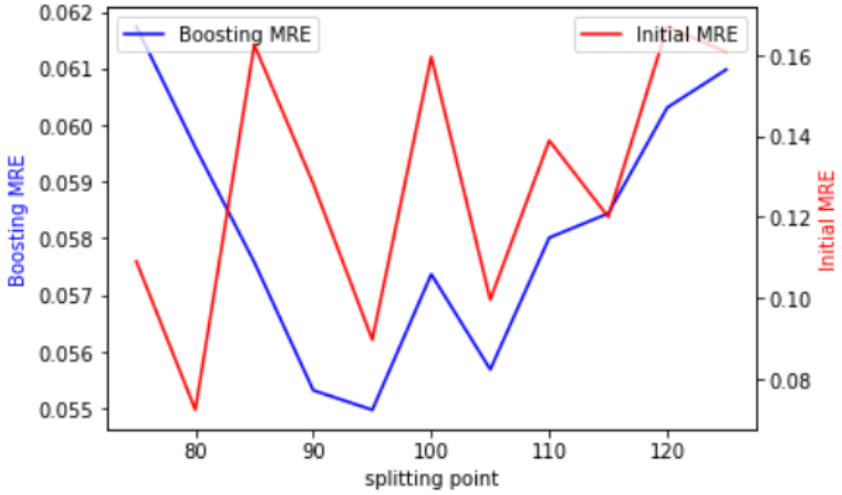

Figure 3: Comparison of Absolute Error between LSTM and Modified LSTM

4. Conclusion and Generalization

This paper employs the LSTM neural network model as the foundation, integrating the model averaging method and sufficient dimensionality reduction theory to predict stock prices. The specific approach involves using the LSTM model for basic modeling, addressing its shortcomings by flexibly applying model averaging to multiple LSTM models with different window widths. Simultaneously, a multifactor model employing sufficient dimensionality reduction is utilized to fit the residual sequence. A comparison with the original LSTM model reveals effective resolution of hyperparameter selection issues and the multifactor model's robust capture of volatility and other crucial information. Particularly, there is a significant improvement in mean square error and absolute error, enhancing the scientific accuracy of experimental results. This demonstrates the superiority and feasibility of the modified LSTM neural network in stock price prediction.

However, due to the stock market's susceptibility to various factors such as political, economic, and psychological influences, the model may not be consistently stable for long-term use. The practicality of the model needs continuous refinement and upgrading to adapt to market fluctuations.

References

[1] White H. Economic Prediction Using Neural Networks: The Case of IBM Daily Stock Returns[C]. Neural Networks IEEE International Conference, 1988, 2(6): 451-458.

[2] Li Chenyang. Research on Stock price prediction and Quantitative stock Selection based on CNN-LSTM [D]. Northwestern University,2021.

[3] Rather A M, Agarwal A, Sastry V N. Recurrent Neural Network and a Hybrid Model for Prediction of Stock Returns [J]. Expert Systems with Applications, 2015, 42(6):3234-3241.

[4] Kristjanpoller W, Fadic A, Minutolo M C. Volatility forecast using hybrid Neural Network models[J]. Expert Systems with Applications, 2014, 41(5):2437–2442.

[5] Patel J, Shah S, Thakkar P, et al. Predicting stock market index using a fusion of machine learning techniques[J]. Expert Systems with Applications An International Journal, 2015, 42(4):2162–2172.

[6] Hochreiter S, Schmidhuber J. Long Short-Term Memory[J]. Neural Computation, 1997,9(8):1735– 1780.

[7] Selvin S, Vinayakumar R, Gopalakrishnan E A. Stock Price Predic-tion Using LSTM, RNN and CNN-sliding Window Model [C]. Interna-tional Conference on Advances in Computing,2017.

[8] Chen Jia, LIU Dongxue, WU Dashuo. Research on Stock Index prediction Method based on Feature Selection and LSTM Model [J]. Computer Engineering and Applications,2019,55(6).

[9] TU Zhirun. Research on Terminal Prediction Model of VOD Refining Furnace based on LSTM [D]. Xi 'an University of Technology,2022.

[10] N Hjort and G Claeskens. Frequentist model average estimators[J]. Journal of the American Statistical Association, 2003 (4):879 − 899.