A Novel Loss Function Considering the Distance Between Forecasting and Historical Values in Financial Time Series Forecasting Models

Weijie Zhang[1,*], Jiaxing Li[2], Yujie Li[3]

{zhangweijie_hit@163.com[1,*], li_jax@outlook.com[2], 18077456639@163.com[3]}

School of Economics and Management, Harbin Institute of Technology, No. 2 West Wenhua Road, Weihai, 264209, China

Abstract. As big data advances by leaps and bounds, a considerable amount of extremely valuable time series data has been accumulated in finance and industry, from which the regularity can be excavated and the future trend can be forecasted. Time series forecasting has obtained widespread application in numerous fields, particularly in financial contexts, and some deep learning methods, including RNN and LSTM, has showed better performance than traditional methods, such as ARIMA, in financial time series forecasting, yet the problem of translation lag in deep learning forecasting still exists. The models may do pseudo learning. In other words, they use the values of the most current historical data directly as the forecasting values. To resolve the issue, this paper designs a new loss function which considers the distance between forecasting and historical values and conducts an experiment on three stock index time series datasets with carrier of the LSTM model. The experimental results reveal that, compared with the traditional MSE loss function, the proposed loss function has higher forecasting accuracy. This research alleviates the translation lag problem triggered by pseudo learning and thus provides a new method for reducing the error of stock index time series forecasting.

Keywords: Stock index forecasting; Long short-term memory; Loss function optimization;

1 Introduction

Along with the development of big data, there has been plenty of data in finance, industry and other fields, of which time series data is an essential component. Time series contains a wealth of information formed as things develops, which is conducive to regularity excavation, phenomena comprehension and future trend forecasting from time series data.

Time series forecasting is a system behavior of forecasting the future based on the current and historical information[1], which has showed enormous values in the fields of temperature forecasting[2], stock index forecasting[3], electricity generation forecasting[4], transportation forecasting[5]. Time series models can date back to 1920s when statisticians utilized traditional measurement models to address time series regression problems. Traditional approaches focus primarily on parametric models informed by domain experts and some of the classic time series analysis models include Autoregressive(AR), Moving Average(MA)[6], Autoregressive Moving Average(ARMA), Autoregressive Integrated Moving Average(ARIMA)[7].

MSIEID 2023, December 08-10, Guangzhou, People's Republic of China

DOI 10.4108/eai.8-12-2023.2344823

With the improvement of data availability and computing power in recent times, machine learning has played a key role in the time series forecasting models of the next generation. Modern machine learning methods has provided an approach to temporal dynamics learning in a purely data-driven manner[8]. Over the past few years, it has become possible to establish deeper models. Compared with the shallow network engineering, they manifest an obvious improvement in learning capacity. Until now, deep learning has produced a great many results in natural language processing, machine translation, voice recognition and other relevant domains. Deep learning works in the problem of time series forecasting and a series of models can continually enhance accuracy of time series forecasts, such as a powerful time series data model RNN[9] and LSTM that can handle the problem of long-distance dependency[10]. In the complex challenges, a single model shows a relatively low degree of fit to time series forecasting, and hence hybrid models are employed to further fit, such as CNN-LSTM[11] and Fast RNN[12]. The new models improve the accuracy of forecasting from the prospective of the structure. Nonetheless, they overlook the impact of translation lag problem caused by deep learning models for time series forecasting on the model accuracy. In the process of time series forecasting, the phenomenon of pseudo learning may occur and the model will take the value of the most current historical data directly as the forecasting value, influencing the accuracy of forecasting. Traditional methods transform the input historical data to mitigate the problem of pseudo learning, which, however, are inapplicable to deep learning models. In an effort to cope with the problem of pseudo learning, this paper proposes a new loss function that considers the distance between forecasting and historical values and evaluates it on three stock index time series datasets. The results indicate that the model has a higher forecasting accuracy than traditional methods and alleviates the problem of pseudo learning to some extent.

In section 2, some concepts relevant to the experiment and computing methods of the new loss function are introduced. In section 3, three stock index time series datasets are used to conduct the experiment and the performance of the new loss function is evaluated. In section 4, the full paper is summarized.

2 Research Methods

2.1 Research Overview

The main overview of the study is illustrated in Fig.1. The stock index data is taken from Investing.com as the time series data and then the data is processed, including normalization and sliding window processing. Normalization can convert data to the same scale for better application to model training and sliding window processing can convert time series data to the input X and the forecasting target value y to train the LSTM model. The input X is selectively logarized to from a logarithmically processed dataset and an unlogarithmically processed dataset and then the data is divided into training set, validation set and test set. Following that, MSE, loss_α, loss_β and loss_γ are employed as the loss functions of the LSTM model. Tuning is performed on the validation set to select the optimal model parameters and the final forecasting result is obtained on the test set. The following subsection outlines the LSTM model, the four loss functions (i.e. MSE, loss_α, loss_β and loss_γ) and the evaluation indicators used in this study.

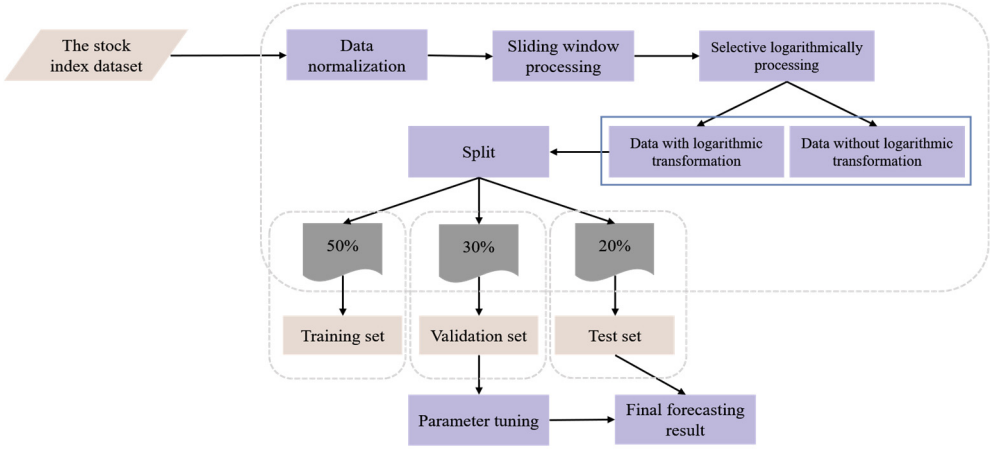

Fig. 1. The flowchart of this study

2.2 Long Short-Term Memory

Long Short-Term Memory (LSTM) is a kind of recurrent neural network (RNN) for processing sequential data. The core idea of LSTM is to introduce a structure called gate to control the flow of information. To be specific, LSTM includes an input gate, a forget gate and an output gate. The input gate determines to what extent the currently input information is incorporated into the memory cells, thereby controlling the storage of new information. The forget date determines what information in the memory cells should be kept or forgotten for the purpose of making room for new information. The output gate determines how to generate the final output according to the current input and the state of the memory cells. Additionally, LSTM has a cell state for storing long-term information. In every time step of LSTM, a new hidden state and a new cell state can be obtained after a series of calculations based on the input vector and the hidden state and the cell state in the last time step. Behind that, the input gate controls the importance of the input vector. The forget gate controls the importance of the cell state and the output gate controls the output of the hidden state. Compared with the traditional RNN, LSTM can avoid the problems of vanishing gradient and exploding gradient more effectively when handling long sequences and is more superior in capturing the long-term dependency relationships.

2.3 Loss Function

MSE. Mean-Square Error (MSE) Loss is also known as L2 Paradigm Loss, which calculates the average of squared differences between actual and forecasting values. The formula, represented as equation (1), is as follows:

$$loss_{MSE} = \frac{1}{N}\sum_{i=1}^{N}\left(\hat{p}_i - p_i\right)^2 \tag{1}$$

where $\hat{P}_i$ is the i-th forecasting stock index value and P_i is the i-th actual stock index value. N represents the number of samples.

The loss function considering the distance between forecasting and historical values. The phenomenon of translation lag results from pseudo learning of the model. The model takes the most recent historical data as the forecasting value. Since the variation of the data between neighboring points in time in time series is not too large in reality, the resultant RMSE is acceptable. Despite that, the accuracy of such a model is low, which is obviously not an expected result.

With the aim of dealing with the problem of translation lag in the results of time series forecasting to a certain degree and thus enhancing the accuracy of the model, this paper proposes a new loss function that takes into account the distance from the forecasting values to the historical values and penalizes the case in which the historical value is directly used as the forecasting value. The three forms of the loss function are expressed as shown in equations (2), (3), and (4).

$$loss_{\alpha} = \frac{1}{N}\sum_{i=1}^{N}\frac{\left(p_i^t - \hat{p}_i^t\right)^2}{\min\left\{\left(\hat{p}_i^t - p_i^{t-2}\right)^2, \left(\hat{p}_i^t - p_i^{t-1}\right)^2\right\}+1} \tag{2}$$

$$loss_{\beta}\frac{1}{N}\sum_{i=1}^{N}\frac{\left|p_i^t - \hat{p}_i^t\right|}{\min\left\{\left|\hat{p}_i^t - p_i^{t-2}\right|, \left|\hat{p}_i^t - p_i^{t-1}\right|\right\}+1} \tag{3}$$

$$loss_{\gamma} = \frac{1}{N}\sum_{i=1}^{N}\left(p_i^t - \hat{p}_i^t\right)^2 + \frac{1}{N}\sum_{i=1}^{N}\lambda\min\left\{\left(\hat{p}_i^t - p_i^{t-2}\right)^2, \left(\hat{p}_i^t - p_i^{t-1}\right)^2\right\} \tag{4}$$

where P_i^t is the actual value of the stock index of the i-th sample at moment t (the value at moment t needs to be forecasted) and P_i^{t-1} is the actual value of the stock index of the i-th sample at moment t-1. $\widehat{P_i^t}$ is the forecasting value of the stock index of the i-th sample at moment t. N denotes the number of samples. λ is the hyper-parameter that represents the weight.

It is worth noting that, for the first two formulas, the difference between the forecasting value and the most recent historical data is introduced into denominator. When the forecasting value is equal to the historical value on the day t-1 or t-2, the value of the loss function becomes larger and the value of the loss function becomes smaller in the opposite cases. Obviously, for the case that the historical value at moment t-1 is equal to that at moment t, $loss_{\alpha}$ is equivalent to MSELoss and $loss_{\beta}$ is equivalent to MAELoss, which are traditional and commonly used loss functions.

For the third formula, the difference between the forecasting value and the historical value is introduced on the basis of MSELoss. λ, a hyper-parameter representing the weight, takes values from [0, 0.01] and is far smaller than the variance of the forecasting and actual values, which enables the model to ignore the error with the historical values and to take the error with the actual values into account.

2.4 Training Method

This study places emphasis on the tuning and optimization of the internal parameters of the model. Through the error back propagation algorithm, loss functions serve to adjust the weights of the paths between neural units in neighboring layers of the model, which can be

translated into tuning of the internal parameters of the model. Consequently, various loss functions will give rise to the difference in the tuning of the internal weight parameters of the model.

As for the artificially set hyper-parameters, such as the number of neural units in each layer, the learning rate, etc., this study tunes the models with different loss functions on the validation set in the experiment and achieves the optimal performance of the model for each loss function under the same framework. After the tuning of the parameters and obtaining the best state of the models, the optimal models for each loss function are adopted and tested on the test set with the aim of comparing the results of the forecasts.

2.5 Evaluation metrics

In order to evaluate the accuracy of the models, three indicators are utilized: Root Mean Squared Error (RMSE), Mean Absolute Percent Error (MAPE), Coefficient of Determination (R^2). The three indicators haven been extensively applied to the evaluation of forecasting models. RMSE and MAPE can be expressed as shown in equations (5) and (6).:

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(\hat{p}_i - p_i\right)^2} \tag{5}$$

$$MAPE = \frac{100\%}{n}\sum_{i=1}^{n}\left|\frac{\hat{p}_i - p_i}{p_i}\right| \tag{6}$$

where $\widehat{P_i}$ is the forecasting value of the i-th stock index and P_i is the actual value of the i-th stock index. N represent the number of data items.

R^2 is the measure of linear correlation between variables and can be calculated as follows in equation(7):

$$R^2 = 1 - \frac{\sum_{i=1}^{N}\left(\hat{p}_i - \overline{p}\right)^2}{\sum_{i=1}^{N}\left(p_i - \overline{p}\right)^2} \tag{7}$$

where the meanings of P_i, $\widehat{P_i}$ and N are the same as those in RMSE. $\overline{P}$ is the mean value of the actual values of the stock indices.

The more R^2 close to 1, the higher the forecasting accuracy. The smaller RMSE and MAPE, the smaller the error.

3 Experiment

3.1 Experimental setting

In this study, three typical and publicly available datasets of stock indices are used: Dow jones Industrial Average(DJIA), Hang Seng Index and FTSE China A50 Index. Data for the three time series indices are obtained from Investing.com. The analysis information of the three datasets is listed in table 1. For DJIA, Hang Seng Index and FTSE China A50 Index, in an effort to provide information for the model as much as possible, historical data including the

closing price, the opening price, the highest price and the lowest price of the past 20 days are input into the model, and the forecasting values for all three datasets are the closing price.

The information of the optimal hyper-parameters for the model using the MSE loss function on Dow jones Industrial Average is presented in table 2. It is a neural network involving a LSTM layer and two fully connected layers. The input layer includes 16×20×4 units and the output layer includes one unit. The first layer is the LSTM layer containing 128 units and the input size is 16×20×4 which can be explained by the batch_size 16, 20 days of historical data and 4 dimensions of the features. The following fully connected layer contains 16 units and the input size is 16×128, meaning that the outputs of the LSTM layer are the inputs of this layer and a 16×16 dimensional output vector is obtained through linear mapping of 16×20×128 features. After that is an activation layer that activates the 16×16 dimensional vector with the ReLU activation function to get an output vector with the same dimension. Finally, there is a fully connected layer that maps the input vector to a scalar, namely the output of the network.

It should be clear that the purpose of this experiment exists in validating the effect of the loss functions instead of finding out the optimal hyper-parameters of the LSTM-based forecasting model for the dataset of this paper. Significantly, for different datasets and different loss functions, the hyper-parameters of the model will be adjusted. The number of units per layer will change, but the number of layers of the model remains the same.

Table 1. The stock index dataset.

Time series	Time range	Train data	Validation data	Test data
DJIA	2013.01.02-2020.12.31	996	598	399
Hang Seng Index	2010.01.04-2022.12.22	996	598	399
FTSE China A50 Index	2013.01.04-2020.12.31	979	588	392

Table 2. The hyper-parameters of LSTM.

Type	Units	Size	Output
LSTM(Input)	128	16×20×4	16×20×128
FC	16	16×128	16×16
ReLU		16×16	16×16
FC(output)	1	16×16	16×1

3.2 Result and Discussion

In order to compare the effect of loss_α, loss_β and loss_γ, this paper used the model with the loss function MSE as the baseline and undertakes the experiment on the three datasets. Moreover, logarithmically processed historical data is introduced as inputs to explore their utilities. Hence there are results of eight models in total.

Table 3 shows the experimental results about using the non-logarithmically processed inputs on the three stock index time series datasets. As for DJIA, it is noticeable that the model using loss_γ as the loss function is the best performer among the four models with non-

logarithmically processed inputs and that the three loss functions proposed in this paper outperform MSEloss on RMSE, MAPE and R^2. On the dataset Hang Seng Index, the model using loss_α shows the best performance and the three loss functions are superior to MSEloss in terms of the model performance. For FTSE China A50 Index, the models using loss_β perform the best and additionally the three models using the proposed loss function still perform better than the model using MSEloss. Above statements fully validate that the proposed loss functions are conducive to a higher accuracy of the model.

Although the proposed loss functions outperform the traditional MSEloss on the three datasets, the best performing loss function on different datasets varies. In other words, none of them can be absolutely better than the rest of the loss functions. The difference between loss_α and loss_β can be understood from the distinction between traditional MAEloss and MSEloss. MAEloss is robust to outliers, while MSEloss is more sensitive to the difference between forecasting and actual values and for MSEloss, a specific analytic solution that converges faster when using gradient descent can be obtained, whereas the difference between loss_γ and the remaining two loss functions can be understood from the mathematical in that multiplication and division bring about different sensitivities.

Table 3. The results without logarithmically processing input

Loss Function	DJIA			Hang Seng Index			FTSE China A50 Index		
	RMSE	MAPE	R^2	RMSE	MAPE	R^2	RMSE	MAPE	R^2
MSE	500.7284	0.0134	0.9407	346.8709	0.0101	0.9513	284.9964	0.0133	0.9450
loss_α	437.1530	0.0110	0.9548	333.9303	0.0097	0.9549	186.7635	0.0090	0.9764
loss_β	454.2692	0.0124	0.9512	335.1524	0.0097	0.9546	184.6012	0.0089	0.9769
loss_γ	436.0320	0.0109	0.9551	336.5541	0.0097	0.9542	189.2545	0.0094	0.9757

Table 4 lists the results with logarithmically processing inputs on the three stock index time series datasets. As for the dataset DJIA, the model utilizing loss_γ perform the best among the four models with logarithmically processing inputs and the three loss functions proposed in the paper are superior to MSEloss on RMSE, MAPE and R^2. On the dataset Hang Seng Index, the model using loss_α is the best performer and the three loss functions proposed in the paper are superior to MSEloss on RMSE, MAPE and R^2. For the dataset FTSE China A50 Index, the model using loss_β shows the best performance and among the models with logarithmically processing inputs, only loss_α and loss_β outperform MSEloss, while loss_γ perform worse than MSEloss, which may be explained by the different utilities (degree of influence) of the logarithmically processing inputs on MSEloss and the new proposed loss functions.

Table 4. The results with logarithmically processing input

Loss Function	DJIA			Hang Seng Index			FTSE China A50 Index		
	RMSE	MAPE	R^2	RMSE	MAPE	R^2	RMSE	MAPE	R^2
ln&**MSE**	460.2149	0.0117	0.9499	340.5883	0.0100	0.9531	199.2089	0.0097	0.9731
ln&**loss_α**	442.9675	0.0109	0.9536	334.6221	0.0096	0.9547	189.9312	0.0092	0.9756
ln&**loss_β**	435.0793	0.0109	0.9553	336.7311	0.0098	0.9541	174.5942	0.0085	0.9794
ln&**loss_γ**	434.5472	0.0110	0.9554	335.2438	0.0097	0.9545	232.6241	0.0112	0.9633

The effects of logarithmically processing inputs on various loss functions are illustrated in Fig.2. In the subgraph(a), for MSEloss, loss_β and loss_γ, logarithmically processing inputs improve the accuracy of the model, but the error of the model using loss_α increases after that. In the subgraph(b), logarithmically processing works for the models using MSEloss and loss_γ, but it results in slight decrease of the performance of the models using loss_α and loss_β. In the subgraph(c), logarithmically processing inputs exerts an adverse influence on the models using loss_α and loss_β and only reduce the error of the models using MSEloss and loss_β.

Through horizontal comparative analysis, it can be summarized that for the three datasets, logarithmically processing inputs invariably enhances accuracy for MSEloss, but always negatively affect loss_α and the effects on loss_β and loss_γ are oscillating and unstable. It validates that logarithmically processing inputs has different effects on the models using different loss functions, which might be caused by the interaction of the newly proposed loss functions and logarithmically processing inputs. In a nutshell, logarithmically processing is less effective for the newly proposed loss functions and even induces decrease in accuracy, which may be clarified by the similar mechanism of logarithmically processing inputs and the newly proposed loss functions and by the duplicated and even conflicting utilities of them.

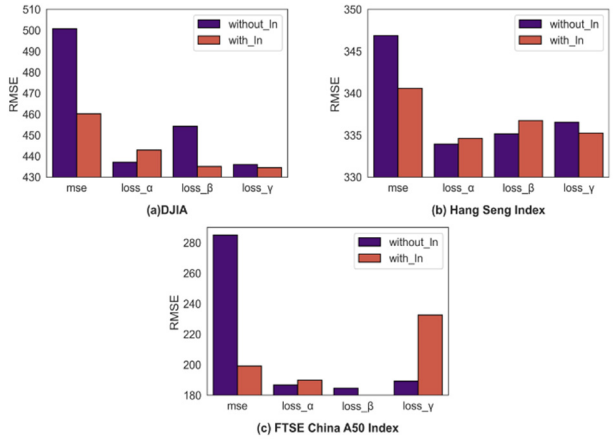

Fig. 2. The comparison of whether to logarithmically process inputs on three datasets.

Regarding the optimal results of the four models with non-logarithmically processing inputs on the three datasets, the average values of R^2 are 0.9457, 0.9620, 0.9609 and 0.9617 respectively, which shows the superiority of the newly proposed loss functions over MSEloss. With respect to the optimal results of the four models with logarithmically processing inputs on the three datasets, the average values of R^2 are 0.9587, 0.9613, 0.9629 and 0.9577, which unveils that the newly proposed loss functions are superior to MSEloss as well.

4 Conclusion

Aiming at the translation lag of the results of stock index time series forecasting, this paper proposes a new loss function that considers the distance between the forecasting and historical

values and validate it on three datasets Dow jones Industrial Average(DJIA), Hang Seng Index and FTSE China A50 Index. The experimental results suggest that, on the three datasets used in this paper, the newly proposed loss function has smaller error and higher accuracy. Additionally, this paper explored the effects of logarithmically processing inputs on the forecasting results and then draws the conclusion that logarithmically processing inputs is partly effective for MSEloss but unstable for the results of the new loss function, which may be explained by the conflicting utilities. Despite the fact that more validation is needed on the applicability of the new loss function, this paper undoubtedly provides a new way for improving the accuracy of stock index time series forecasting.

Acknowledgments. The authors thank the Laboratory of Finance and Information Technology at Harbin Institute of Technology, Weihai, for their invaluable support and assistance throughout this research endeavor. The expertise and guidance provided by the laboratory played an indispensable role in the successful completion of this work.

References

[1] De Gooijer, J.G., Hyndman, R.J., 25 years of time series forecasting, Int. J. Forecast. 22(3) (2006) 443-473.

[2] Hewage, P., Behera, A., Trovati, M. et al. Temporal convolutional neural (TCN) network for an effective weather forecasting using time-series data from the local weather station. Soft Comput 24, 16453–16482 (2020)

[3] Y. Kaneko, "A Time-series Analysis of How Google Trends Searches Affect Cryptocurrency Prices for Decentralized Finance and Non-Fungible Tokens," 2021 International Conference on Data Mining Workshop

[4] R. Palma-Behnke, F. Valencia, J. Vega-Herrera and O. Núñez-Mata, "Synthetic Time Series Generation Model for Analysis of Power System Operation and Expansion with High Renewable Energy Penetration," in Journal of Modern Power Systems and Clean Energy, vol. 9, no. 4, pp. 849-858, July 2021.

[5] C. Ma, G. Dai and J. Zhou, "Short-Term Traffic Flow Prediction for Urban Road Sections Based on Time Series Analysis and LSTM_BILSTM Method," in IEEE Transactions on Intelligent Transportation Systems, vol. 23, no. 6, pp. 5615-5624, June 2022.

[6] Box GEP, Jenkins GM. 1976Time series analysis: forecasting and control. San Francisco, CA: Holden-Day. Google Scholar

[7] Box, G.E.P.; Pierce, D.A. Distribution of residual autocorrelations in autoregressive-integrated moving average time series models.

[8] Ahmed NK, Atiya AF, Gayar NE, El-Shishiny H. 2010An empirical comparison of machine learning models for time series forecasting. Econ. Rev. 29, 594–621.

[9] A.Graves, A. -r. Mohamed and G. Hinton, "Speech recognition with deep recurrent neural networks," 2013 IEEE International Conference on Acoustics, Speech and Signal Processing, Vancouver, BC, Canada, 2013, pp. 6645-6649.

[10] Hochreiter, Sepp and Jürgen Schmidhuber. “Long Short-Term Memory.” Neural Computation 9 (1997): 1735-1780.

[11] Zhang, Jiaxuan and Shun Li. “Air quality index forecast in Beijing based on CNN-LSTM multi-model.” Chemosphere (2022): 136180 .

[12] M. A. Istiake Sunny, M. M. S. Maswood and A. G. Alharbi, "Deep Learning-Based Stock Price Prediction Using LSTM and Bi-Directional LSTM Model," 2020 2nd Novel Intelligent and Leading Emerging Sciences Conference (NILES), Giza, Egypt, 2020, pp. 87-92.