Prediction of Index Trend based on LSTM Model for Extracting Image Similarity Feature

Junming Guo†
Sichuan University
Chengdu, China
junming_guo@foxmail.com

Xuwei Li
Sichuan University
Chengdu, China
lixuwei@scu.edu.cn

ABSTRACT

The prediction of stock trend in the financial market has become an important application field in machine learning research. The fluctuation of stock price is affected by various factors and strong nonlinear and stochastic, which brings greater challenge to predict the future trend of stock. This paper demonstrates how to predict the index trend more accurately so as to bring more profits to investors. A large number of researchers in the financial field have tried to use machine learning algorithms, such as Support Vector Machines (SVM), Random Forests (RF), and Deep Neural Network (DNN) like Convolutional Neural Network (CNN) and Long Short Term Memory (LSTM) Network models to seek the dependence on the future trend of stock and historical stock data. This paper attempts to use Convolutional Autoencoder (CAE) to extracts the image implicit feature of the graphical representation to index data, and calculate the cosine similarity between other implicit features of its historical images. The image similarity feature is equal to the total sum of cosine similarity products by multiplying the corresponding trend direction value. Finally, we combine the image similarity feature and the filtered data features by Pearson product-moment correlation coefficient (PPMCC) as the input variables of the LSTM network model. After discussing the results of the comparative experiments, the prediction model with incorporating the image similarity feature has been improved on multiple classification evaluations.

CCS CONCEPTS

• **Human-centered computing ~ Collaborative and social computing** • Human-centered computing ~ Social engineering (social sciences)

KEYWORDS

Image feature extraction, Long short term memory network, Trend prediction, Financial time series

1 Introduction

In the macroeconomic environment, financial markets are dynamic systems with strong randomness, complexity, and non-linearity. Stock trading is one of the most typical behaviors in investment financial markets. The fluctuation of stock prices is closely related to some multifaceted and uncertain factors such as financial policies, performance of listed companies, industry performance, investor sentiment, and other economic factors. Financial time series are highly volatile and non-stationary, so predicting financial time series has been proved to be a challenging task [1]. Researchers in the financial field mainly used time series, machine learning and deep learning algorithms to model and analyze stock historical data, combining the knowledge of probability and statistics to construct a stock price forecasting model. For example, ZHANG et al. [2]used a linear method to build ARIMA model to predict the price of CSI 300 index; Chen et al. [3] proposed a feature-weighted SVM combined with the K-nearest neighbor algorithm to predict the index price by optimizing the stock data feature weights; Khaidem et al. [4] attempted to exponentially smooth stock historical data, using the Random Forest (RF) model on the integrated learning approach to predict the future k-day stock price trend. Traditional machine learning algorithms are difficult to ensure better fitting of nonlinear data. However, neural network can make up for the poor nonlinear fitting of machine learning traditional algorithms. Therefore, many scholars obtained prediction results with higher precision using deep learning algorithms for application of stock price prediction in recent years. Kulaglic et al. [5] utilized discrete wavelet transform (DWT) as well as two types of neural network (NN) models to predict the stock price. Tsantekidis et al. [6] analyzed stock trade order history data and used Convolutional Neural Network to extract implicit features to predict stock price trend; Yao et al. [7] used Long Short Term Memory modeling to predict the future trend of CSI 300 index. Some researchers began to consider the characteristics of the data on stock prices to diversify expression, such as Hu et al. [8] using candlesticks to represent stock price changes and obtaining lower dimensional implied features by Convolutional Autoencoder. It can categorize stocks by combining the implied features and network modular clustering, and optimize the investment portfolio calculating the Sharpe ratio of each class of stock portfolio. Udagawa [9] reconstructs the feature representation of the stock price candle chart to achieve short-term price prediction for the S&P 500 and Nasdaq index data.

AICS 2019, July 12–13, 2019, Wuhan, Hubei, China

ACM ISBN 978-1-4503-7150-6/19/07…$15.00
https://doi.org/10.1145/3349341.3349427

In this paper, the index trend prediction research, the image feature is generated by combined with the convolutional autoencoder and the cosine similarity, and it multiplies with the corresponding trend direction value to obtain the similarity feature. The Pearson correlation coefficient is used to select the important features of the original data and common technical indicators of the index. The LSTM model was constructed to predict the trend of index by taking the image similarity feature and the important features as variables features. In order to verify the effectiveness and stability of the image similarity feature, this paper selects the historical data of 4 indexes published by Tushare for comparison experiments. The experimental results show that the LSTM model combined with the image similarity feature gets better in many evaluating indicators.

Our contributions in this work include: 1) using Convolutional Autoencoder to capture the implicit feature from the graphical representation of index data; 2) calculating a novel important feature by measuring the similarity among index historical data based on cosine similarity; 3) introducing LSTM network model architecture and predicting the trend of index with the technical indicator and novel feature.

The rest of the paper is organized as follows. Section 2 introduces the relevant methods involved in the model. Section 3 describes the construction process of the model proposed in this paper. Experimental data, parameter selection, and results are shown in Section 4. In the end, we make a conclusion about our work in Section 5.

2 Related Work

This section mainly describes the related methods involved in the model, including the graphical representation of index data, convolutional autoencoder, selection of relevant features of stock data, and long short term memory network.

2.1 Index Data Representation

The original index data uses daily data, which includes the opening price, the lowest price, the highest price, the closing price, and the transaction volume of the day. Visualization can be chosen line chart, histogram, scatter plot, etc. For the index price data, the candlestick chart is usually used to represent it. For instance, Hu et al. [8] used candlesticks to represent stock price data and clustered the generated candlesticks into a portfolio. This paper uses the candlestick chart to graphically represent the index price daily data, as is shown in figure 1:

Figure 1: Candlestick chart of index price daily data

2.2 Convolutional Autoencoder

Autoencoder is an unsupervised learning algorithm in the deep learning model. It abstracts features layer by layer by imitating human brain through neural network and reconstructs input data representation with sparse feature vector. Masci et al. [10] had indicated that a fully connected autoencoder will ignore 2D image structure information, resulting in all features of the graph belonging to global variables, and generate a large number of redundant parameters. The convolution autoencoder can be fitted the problem by sharing network weights and the local features of the image are preserved. The convolution autoencoder is used to generate the sparse matrix features, and the convolutional decoder is used to optimize the network parameters, and its structure is shown in figure 2:

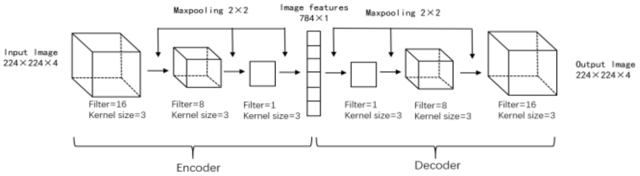

Figure 2: Convolutional autoencoder and decoder

The convolutional autoencoding extracts the image implied features from the candlestick chart of index price daily data in Section 2.1 as an input image.

2.3 Cosine Similarity

Cosine similarity is defined as the degree of similarity between two vectors by calculating the cosine of the angle between them. It is applied widely in text classification, document clustering and face recognition [11]. The calculation formula is as follows:

$$Cosine\ Similarity = \frac{A \bullet B}{\| A \| \| B \|} = \frac{\sum_{m=1}^{N} A_m \times B_m}{\sqrt{\sum_{m=1}^{N} (A_i)^2} \times \sqrt{\sum_{m=1}^{N} (B_i)^2}} \tag{1}$$

Where A_m and B_m represent the mth component of vectors A and B, and N is the length of the vector. The degree of similarity is according to the cosine similarity between the implied feature of the images among the historical data, and is normalized by $softmax$ function which is defined as:

$$softmax(cs_{ij}) = \frac{e^{cs_{ij}}}{\sum_{k=1}^{K} e^{cs_{ik}}} \quad s.t.: i \neq j \tag{2}$$

Where cs_{ij} represents the cosine similarity between the ith and jth image implied feature, and K represents the total number of historical data images.

2.4 Feature Selection of Index Data

Feature subset selections is an important problem that has many ramifications [12]. The original index data includes the opening price, the lowest price, the highest price, the closing price and the transaction volume. Researchers usually consider the calculation of common technical indicators when analyzing stock data in the financial field. However, it may lead the prediction model overfitting while taking all the technical indicators as model input, so it is necessary to select significant features to optimize the model.

2.4.1 Technical Indicators. The financial technical indicators are based on some complicated calculations and certain mathematical statistics methods. It can reflect the deeper connotation of the original stock data which has important reference value for predicting the future trend of stock price. This paper considers the closing price, the lowest price, the highest price, the closing price, the transaction volume of the original index data and 16 technical indicators as the feature selection of the index data. The part of technical indicator can choose different parameter (Range of days) as is shown in table 1:

Table 1: Technical Indicator

Technical indicator name	Abbreviation	Parameter
Simple moving average	SMA	5/10/15
Williams %R	WR	14
Relative Strength Index	RSI	15/20
Ultimate Oscillator	UOS	7/8/9
KDJ Index	KDJ	
Moving Average Convergence	MACD	9/10
On Balance Volume	OBV	
Money flows indicators	MFI	14/18
Rate of Change	ROCP	

Data normalization processing is the basic work of data mining in machine learning and deep learning. Different data features often have different dimensional units. Therefore, it is necessary to normalize the data to modeling on all kinds of data features in the same order of magnitude. We use the Min-Max Normalization (MMN), also known as linear normalization or deviation normalization, linearly change to a value between 0 and 1. Above five original index data features and 16 technical indicators are normalized by MMN, and the values of features on the training set are standardized by:

$$\tilde{x}_t = \frac{x_t - x_{min}}{x_{max} - x_{min}} \tag{3}$$

Where x_{min} and x_{max} are the minimum and maximum value of x in the training set.

2.4.2 Feature Selection. This paper mainly forecasts and analyzes the trend of the index closing price, so the index trend value of the $\boldsymbol{tth}$ day index is calculated as follows:

$$Trend_t = Close_{t+1} - Close_t \tag{4}$$

Where $Close_t$ represent the closing price of the tth day, $Trend_t$ is a positive number represent a rise, and vice versa is a falling or flat, and the index trend direction value is defined as follows:

$$Trend\ direction_t = \begin{cases} 1 & if\ Trend_t > 0 \\ -1 & if\ Trend_t \leq 0 \end{cases} \tag{5}$$

Pearson product-moment correlation coefficient (PPMCC) is one of the most common measurement of determining linear dependence, reflecting the degree of linear correlation between two variables [13]. A value greater than 0 indicates a positive correlation. Less than 0 is negatively correlated, and 0 represents linearly independent. The calculation formula is as follows:

$$\rho_{p,q} = \frac{\sum_{t=1}^{D}(x_{p,t} - \bar{x}_p)(x_{q,t} - \bar{x}_q)}{\sqrt{\sum_{t=1}^{D}(x_{p,t} - \bar{x}_p)^2 \sum_{t=1}^{D}(x_{q,t} - \bar{x}_q)^2}} \tag{6}$$

Where $x_{p,t}$ and $x_{q,t}$ represent the value of the pth and the qth feature on the tth day index, $\overline{x_p}$ and $\overline{x_q}$ represents the average value of the pth and qth feature, and the total D day data. This section selects the CSI 300 daily data, and calculates absolute value order of PPMCC between the above 21 data features and trend value, as are shown in figure 3:

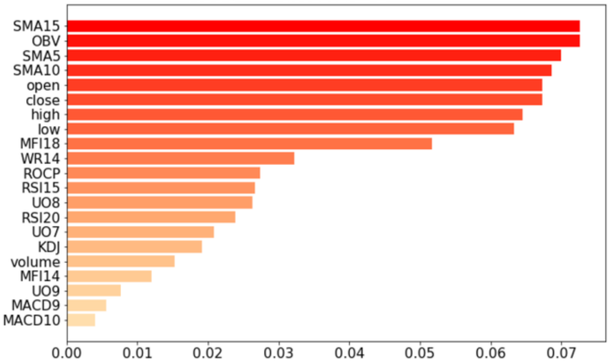

Figure 3: PPMCC between features and trend value

We choose the important feature with the highest PPMCC value among the same technical indicator. 8 features including SMA15, OBV, opening price, closing price, highest price, lowest price, MFI18 and WR14 are selected as the input variables of the prediction model.

2.5 Long Short Term Memory

The traditional neural network model assumes that the input data segments of the network are independent of each other, making it difficult to process time series data. The time series data is usually selected as the Recurrent Neural Network (RNN) in the deep learning algorithm. In the recurrent neural network, the output result of a sequence depends on the current input and the state of the network at the previous moment. Although the network design can memorize the historical data information, the historical data information will be gradually forgotten. Therefore, Hochreiter [14] proposed a Long Short Memory Neural Network (LSTM) to overcome the defect of forgetting information comparing RNN. The storage unit in the LSTM network contains one or more neural units, and the input gate, the forget gate, and the output gate. The gates are information-selective methods that allow selective memory or forgetting of data [15]. Memory unit internal structure of LSTM is shown in figure 4 below:

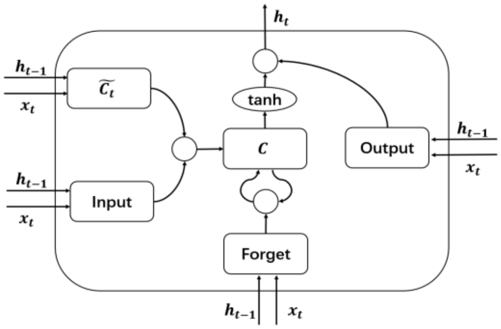

Figure 4: Memory unit internal structure of LSTM

The following equation will describe the update process of the LSTM memory cell at time *tth*, and x_t represents the input of the memory cell at time *t*, and h_t represents the output of the memory cell at time *tth*. W_{xi}、W_{xf}、W_{xc}、W_{xo}、W_{hi}、W_{hf}、W_{hc}、W_{ho} are weight matrix, b_i、b_f、b_c、b_o are offset vector, and σ、*tanh* are activation function

First, based on the input value of the storage unit at the tth time and the output value of the storage unit at the *(t-1)th* time, calculate the input gate, the forget gate, the output gate, and the candidate state value:

$$Input_t = \sigma(W_{xi}x_t + W_{hi}h_{t-1} + b_i) \tag{7}$$

$$forget_t = \sigma(W_{xf}x_t + W_{hf}h_{t-1} + b_f) \tag{8}$$

$$\tilde{C}_t = tanh(W_{xc}x_t + W_{hc}h_{t-1} + b_c) \tag{9}$$

$$output_t = \sigma(W_{xo}x_t + W_{ho}h_{t-1} + b_o) \tag{10}$$

Second, based on the input gate, the candidate state value and the forget gate at the tth time, and the storage unit state value at the (t-1)th time, calculate the storage unit state value at the tth time:

$$C_t = Input_t * \tilde{C}_t + forget * C_{t-1} \tag{11}$$

Thirdly, the output value of the storage unit at tth time is calculated based on the output gate and the storage unit state value at tth time:

$$h_t = output_t * tanh(C_t) \tag{12}$$

The input variables of the LSTM model in this paper are from eight data features in section 2.4. As a two-category prediction, the $softmax$ activation layer is added in front of the output layer in the LSTM model. The output result is the probability of predicting the rise and fall, and the prediction trend with the larger probability is selected as the forecast result. The length of time window can be found in the experimental part of section 4.

3 Lstm Model based on Extracting Image Similarity Feature

The LSTM model based on the extracted image similarity feature proposed in this paper, on the one hand, graphically represents the opening price, the lowest price, the highest price and the closing price in the index historical data, and uses the convolution autoencoder to extract the implicit features of image. Cosine similarity is used to calculate the similarity degree between the implied feature of images and it is normalized by $softmax$ function. The similarity feature is equal to the total sum of product of the index trend direction value and the cosine similarity. The calculation formula is as follows:

$$Similarity\ feature_i = \sum_{i=1}^{K} Trend\ direction_j \times softmax(cs_{ij})\ s.t.: i \neq j \tag{13}$$

On the other hand, the common technical indicators are calculated from original index data, and the PPMCC is used for feature selection. The input variables of LSTM model are as the image similarity feature and the filtered features by PPMCC. The construction process of the prediction model is shown in figure 5:

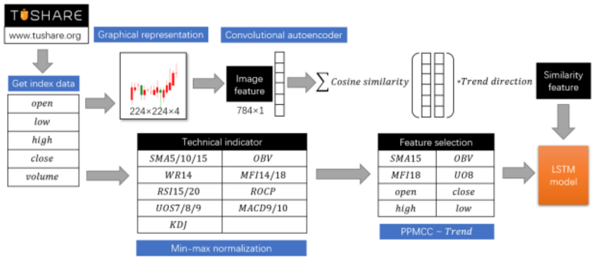

Figure 5: The construction process of prediction model

4 Experiment

This section is to verify the effectiveness and stability of the LSTM model based on the extracted image similarity feature, and introduce the experimental data, the selection of the length of time window in LSTM and experimental results will be mainly introduced.

4.1 Experimental Data

The experimental data is obtained from the financial data provided by Tushare's public data source API interface, including Shanghai composite index (SH), Shenzhen component index (SZ), CSI 300, and Shanghai Stock Exchange 50 index (SSE 50), and selected time range is from February 18, 2016 to December 28, 2018. The trend of index closing price was taken as the evaluation standard, and 80% of the data were divided into training set and the rest of data were testing set.

4.2 Selection of Parameters in lstm

It exists diverse variation rules and data distribution of different index data, so the parameters of the LSTM model required to be adjusted appropriately, such as the length of time window, optimizer, and epochs.

We will discuss the length of time windows from the range of 10 ~ 20 days on 20% of CSI 300 training set as validation set, and obtain the best length of time window by observing the loss curve of validation set on different days:

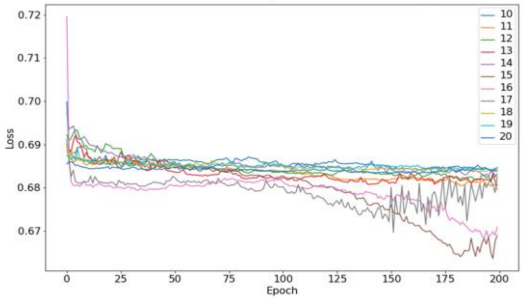

Figure 6: Loss curve of validation set on different days

At the same time, the epochs also should be considered as the important factor in the process of prediction, and the accuracy and loss curve of training and validation set are shown in figure 7:

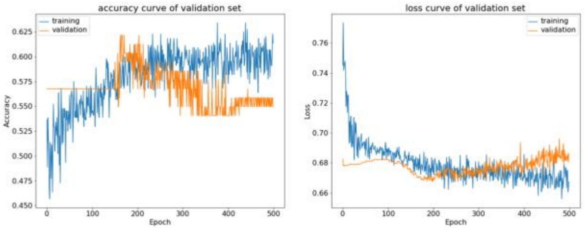

Figure 7: Accuracy and loss curve of training and validation set

We can make a conclusion that the lower loss and volatility of the validation set when 200 epochs from figure 7, and 15 days is selected as the best length of time window on account that the loss arrives low enough of validation data set within the short time comparing other days.

4.3 Experimental Results

In this section, the experimental results will be analyzed from the following four evaluation indicators: Accuracy, Precision, Recall, and F1-measure. The experimental results are shown in table 2:

Table 2: Experimental results

Index	Model	Acc(%)	Prec (%)	Rec(%)	F1 (%)
SH	M1	56.62	47.36	31.03	37.49
	M2	55.89	46.93	39.65	42.99
SZ	M1	58.37	47.36	15.78	23.68
	M2	64.13	65.21	26.31	37.49
CSI 300	M1	59.22	49.99	22.80	31.32
	M2	67.25	68.96	35.08	46.51
SSE 50	M1	57.34	48.64	31.03	37.89
	M2	59.52	52.37	37.93	43.99

Where M1 represents the LSTM model constructed only by eight data features filtered by PPMCC, and M2 represents the LSTM model constructed with image similarity feature and eight features. The time window length of 15 days is selected in both of two models.

By comparing the experimental results, the LSTM prediction model based on the extracted image similarity feature has scored higher on most of various indicators, which can prove that the image similarity feature is able to provide effective information for the prediction model, and optimizing the model performance.

5 Conclusion

In the financial field, a large number of researchers use the statistical analysis of financial historical data to explore implied information, and predict future trends for maximizing returns. This paper extracts the images feature by using cosine similarity and convolutional autoencoding from the graphical representation of the index price daily data. The total sum of multiplying the corresponding trend direction value can generate the image similarity feature. We construct the LSTM model with the image similarity and eight filtered important features by PPMCC to predict the future trend of the index. Comparing the experimental results, the prediction model with image similarity feature can get better performance.

ACKNOWLEDGMENTS

I would like to express my appreciation of the help from my schoolmates, who have provided me with valuable suggestions during the writing of this paper.

REFERENCES

[1] Wen M, Li P, Zhang L, et al. Stock Market Trend Prediction Using High-order Information of Time Series [J]. IEEE Access, 2019.

[2] ZHANG Y A T, SUN B O. Analysis of CSI 300 Stock Index Futures Price Trend Based on ARIMA Model [J]. DEStech Transactions on Social Science, Education and Human Science, 2017 (seme).

[3] Chen Y, Hao Y. A feature weighted support vector machine and K-nearest neighbor algorithm for stock market indices prediction[J]. Expert Systems with Applications, 2017, 80: 340-355.

[4] Khaidem L, Saha S, Dey S R. Predicting the direction of stock market prices using random forest [J]. arXiv preprint arXiv:1605.00003, 2016.

[5] Kulaglic A, Üstündağ B B. Stock Price Forecast using Wavelet Transformations in Multiple Time Windows and Neural Networks [C]//2018 3rd International Conference on Computer Science and Engineering (UBMK). IEEE, 2018: 518-521.

[6] Tsantekidis A, Passalis N, Tefas A, et al. Forecasting stock prices from the limit order book using convolutional neural networks [C]//2017 IEEE 19th Conference on Business Informatics (CBI). IEEE, 2017, 1: 7-12.

[7] Yao S, Luo L, Peng H. High-Frequency Stock Trend Forecast Using LSTM Model[C]//2018 13th International Conference on Computer Science & Education (ICCSE). IEEE, 2018: 1-4.

[8] Hu G, Hu Y, Yang K, et al. Deep Stock Representation Learning: From Candlestick Charts to Investment Decisions[C]//2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2018: 2706-2710.

[9] Udagawa Y. Predicting Stock Price Trend Using Candlestick Chart Blending Technique[C]//2018 IEEE International Conference on Big Data (Big Data). IEEE, 2018: 4162-4168.

[10] Masci J, Meier U, Cireşan D, et al. Stacked convolutional auto-encoders for hierarchical feature extraction[C]//International Conference on Artificial Neural Networks. Springer, Berlin, Heidelberg, 2011: 52-59.

[11] Pieterse J, Mocanu D C. Evolving and Understanding Sparse Deep Neural Networks using Cosine Similarity[J]. arXiv preprint arXiv:1903.07138, 2019.

[12] John G H, Kohavi R, Pfleger K. Irrelevant features and the subset selection problem[M]//Machine Learning Proceedings 1994. Morgan Kaufmann, 1994: 121-129.

[13] Puth M T, Markus Neuhäuser, Ruxton G D. Effective use of Pearson's product-moment correlation coefficient Comment [J]. Animal Behaviour, 2014, 93:183-189.

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

[15] Liu S, Liao G, Ding Y. Stock transaction prediction modeling and analysis based on LSTM[C]//2018 13th IEEE Conference on Industrial Electronics and Applications (ICIEA). IEEE, 2018: 2787-2790.