Stock Market Prediction-by-Prediction Based on Autoencoder Long Short-Term Memory Networks

Conference Paper · December 2020

DOI: 10.1109/ICEE50131.2020.9261055

3 authors:

Mehrnaz Faraz
Khaje Nasir Toosi University of Technology

Hamid Khaloozadeh
Khaje Nasir Toosi University of Technology

Milad Abbasi
University of Alberta

Stock Market Prediction-by-Prediction Based on Autoencoder Long Short-Term Memory Networks

1st Mehrnaz Faraz
Department of Electrical Engineering
K. N. Toosi University of Technology
Tehran, Iran
mehrnazfaraz@email.kntu.ac.ir

2nd Hamid Khaloozadeh
Department of Electrical Engineering
K. N. Toosi University of Technology
Tehran, Iran
h_khaloozadeh@kntu.ac.ir

3rd Milad Abbasi
Department of Electrical Engineering
Sharif University of Technology
Tehran, Iran
miladabbasi2012@gmail.com

***Abstract*—In this paper, we address the prediction-by-prediction of the stock market closing price using the autoencoder long short-term memory (AE-LSTM) networks. To integrate technical analysis with deep learning methods, technical indicators and oscillators are added to the raw dataset as features. The wavelet transformation is used as a noise-removal technique in the stock index. Anomaly detection in dataset is also carried out by means of the z-score method. First, the autoencoder is trained to represent the data. Then, the encoder extracts feature and puts them into the LSTM network for predicting the closing price of the stock index. Afterwards, the system predicts subsequently based on the previous predictions. We choose the standard and poor's 500 (S&P 500) stock market index to assess the proposed method through several simulation studies. Our performance criteria include mean absolute error (MAE), root mean square error (RMSE), and average return (AR). The results show that the proposed prediction-by-prediction method outperforms generative adversarial network (GAN) in terms of daily adjusted closing price prediction.**

***Index Terms*—Autoencoder, deep learning, feature extraction, long short-term memory, prediction-by-prediction, stock market prediction, wavelet transformation.**

I. Introduction

The stock price prediction is extremely challenging because of the random, nonlinear, nonstationary, and noisy behavior of stock markets. In addition, highly interconnected factors such as economic, political, and psychological variables have effects on the stock markets.

In the last few decades, many financial corporations use artificial intelligence (AI) methods in stock market prediction. Among the AI techniques, deep learning has become a dominant and popular tool in the financial market analysis and has had many promising results in stock price forecasting through nonlinear, data-driven, and multivariate analysis [1]–[3].

Various deep artificial neural networks (ANNs) have attracted the researchers' attention in time series prediction [4]. In [5], a hybrid deep ANN model composed of long short-term memory (LSTM) and gated recurrent unit (GRU) is proposed and the prediction model is tested using S&P 500 time series. The hybrid approach with combined sentiment analysis and deep convolutional neural network (CNN) for the prediction of S&P 500 index is also discussed in [2] to analyze the short and long-term influences of news events on the index. An ensemble of deep ANNs to predict Chinese stock markets is also proposed in [6]. Recurrent neural networks (RNNs) [7], LSTMs [8], deep belief networks (DBNs) [9], [10], and reinforcement learning (RL) [11] methods are also used in stock price prediction. The stock market prediction based on generative adversarial networks (GANs) is also discussed in [12].

The two major approaches to make decisions in financial markets are fundamental and technical analysis. Technical analysis is based on the direct patterns in stock data and visual aspects of charts to derive indicators such as moving average (MA), relative strength index (RSI), stochastic oscillator, etc. [13]. On the other hand, fundamental analysis is the process of analyzing the stock market at the basic, fundamental financial level to measure its intrinsic value [14].

Most researchers have focused on deep learning frameworks that rely on the raw dataset or a limited number of features [15]–[17]. Regarding the complexity of financial time series, we have combined deep learning with technical analysis; therefore, technical indicators and macroeconomic conditions are fed to the system as a multivariate signal to forecast future stock prices.

In this paper, a novel daily stock price prediction-by-prediction model based on autoencoder long short term memory (AE-LSTM) with the LSTM network as the forecaster and AE as the feature extraction is proposed. Autoencoder, as the main part of our model, learns the invariant and abstract features in an unsupervised way [18]. To eliminate the market noise and get the main stock trend, wavelet transformation is used. Also, the outliers in data are excluded before training the network. The main contribution of this paper is to design an encoder-decoder based model to represent daily features and predict the financial time series using the encoded representation of data. In addition, a state-of-the-art, complete set of modules for denoising, removing outliers, considering technical indicators, and extracting deep feature instead of feature selection is used.

This paper is organized as follows: in Section II, we describe the deep learning models which are used in our analysis. We describe details of the dataset as well as a detailed description of the features used in our experiments in Section III. In Section IV, we provide the experimental protocol and results. Finally, Section V concludes the paper.

II. Prediction Models

A. Long Short-Term Memory Networks

LSTM networks are a modified version of recurrent neural networks (RNNs), which remembers the long term dependencies in data in an efficient way. RNNs encounter the vanishing gradient problem, while this problem is resolved in LSTM networks.

The core of an LSTM network is a memory unit (or cell) which is shown in Fig. 1. A cell consists of three sigmoid and one tanh layers, which form three gates organizing the information inside and outside of the cell. The input and output gates control the input and output information in the memory unit, respectively. The forget gate can reset the memory unit with a sigmoid function.

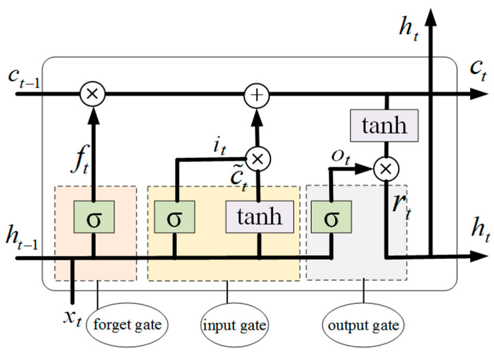

Fig. 1. The architecture of an LSTM cell.

Given the information x_t, the flow of information in an LSTM cell can be formulated as follows:

$$f_t = \sigma\left(W_f . \left[h_{t-1}, x_t\right] + b_f\right) \tag{1}$$

$$i_t = \sigma\left(W_i . \left[h_{t-1}, x_t\right] + b_i\right) \tag{2}$$

$$\tilde{C}_t = \tanh\left(W_C . \left[h_{t-1}, x_t\right] + b_C\right) \tag{3}$$

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \tag{4}$$

$$o_t = \sigma\left(W_o \left[h_{t-1}, x_t\right] + b_o\right) \tag{5}$$

$$h_t = o_t * \tanh\left(C_t\right) \tag{6}$$

where f_t, i_t, and o_t indicate the forget, input, and output gates at time t, respectively. C_t is the cell state vector, which is updated in (4), and h_t represents the hidden state at the current time t. Tanh is the hyperbolic tangent function, and $*$ in (4) and (6) denotes a point-wise multiplication operator. The new cell state (C_t) which represents the new information, scaled by how much we decided to update each state value.

B. Autoencoder

An autoencoder is a ANN that includes two parts: an encoder $h = f(x)$ and a decoder that produces a reconstruction $\hat{x} = g(h)$. The purpose of designing autoencoders is to copy the input of the network to its output. The model is forced to give importance to the useful aspects of input; in other words, an autoencoder learns useful properties of the data. Therefore, these networks do not learn to copy perfectly, but approximately, and they are restricted in such a way that generate output that resembles the training data, as shown in Fig. 2.

C. Autoencoder LSTM

In this paper, we use autoencoder to compress and encode the data. Autoencoder is an unsupervised ANN that prepares a reduced encoded representation of data and then learns how to reconstruct the data back from them. The similarity of the reconstructed data to the original input determines the effectiveness of an autoencoder. The forecasting workflow used in this paper is shown in Fig. 3. The autoencoder consists of LSTM layers in encoder and decoder parts, and after each LSTM layer, we use dropout as a regularization method to prevent from overfitting. First, the autoencoder is trained. Next, the encoder part is used as the feature generator. The final step is to train the LSTM based forecaster, so that the adjusted closing price prediction of next day is provided.

We define w as the time step in time series data and use $x_t, x_{t+1}, \cdots, x_{t+w}$ to predict the adjusted closing price at the next day. $\mathbf{X}$ is the input of the AE-LSTM network, which is defined as follows:

$$\mathbf{X} = \{x_t, x_{t+1}, ..., x_{t+w}\} \tag{7}$$

Note that the real adjusted closing price at the next day, i. e. x_{t+w+1}, is used as the target in the training phase.

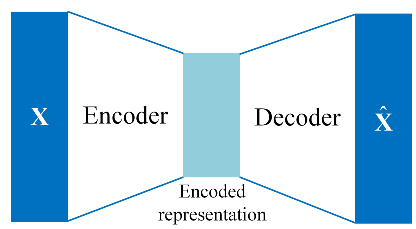

Fig. 2. The architecture of an Autoencoder.

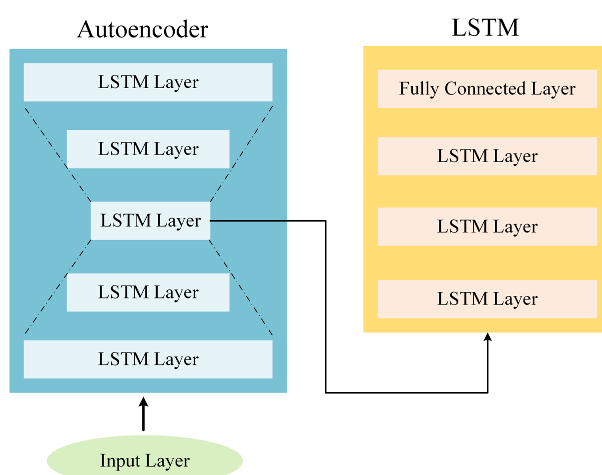

Fig. 3. The architecture of Autoencoder LSTM.

III. Dataset Analysis and Descriptions

In this part, we discuss briefly the methodology of dataset analysis. The details of each step will be discussed in this section, as follows:

A. Dataset Description

We evaluate our model on the S&P 500 Index stock data. The historical data can be downloaded in Yahoo Finance. We select the date frame from 1/3/2000 to 11/4/2019 (almost 5000 pieces of data). The raw data consists of six features: open, high, low, closing, and adjusted closing prices, as well as the volume of trading. Open, high, low, and closing prices are the first, highest, lowest, and the last stock prices exchanged on a specific trading day, respectively. We generate nine technical indicators and add them to the dataset to incorporate technical analysis into our prediction. Technical indicators are important parameters that are derived from stock data to predict the movement of the financial market. In the following, we briefly define and formulate these technical indicators:

- Simple Moving Average (SMA)
  SMA represents the moving market average for a given period. The SMA is calculated as follows:

$$SMA(C,5) = \sum_{k=a+1}^{a+5} \frac{C\left(k\right)}{5} \tag{8}$$

  where C is the closing price.
- Moving Average Convergence Divergence (MACD)
  MACD is a technical indicator showing the trend of the stock prices which is calculated by subtracting the 26-period exponential moving average (EMA) from the 12-period EMA as follows:

$$MACD = EMA(C,t,12) - EMA(C,t,26) \tag{9}$$

  where EMA is a type of SAM that signifies the most recent data points. The $EMA(C,t,\tau)$ shows the τ day EMA and can be obtained as follows:

$$\begin{aligned} EMA(C,t,\tau) = C\left(t\right) - EMA\left(C,t-1,\tau\right).\frac{2}{\tau+1} \\ + EMA\left(C,t-1,\tau\right) \end{aligned} \tag{10}$$

- Relative Strength Index (RSI)
  RSI is an index of oscillator-type technical analysis that shows the historical strength and weakness of stock price. It is formulated as follows:

$$RSI = 100 - \frac{100}{1+\frac{average\ gain}{average\ loss}} \tag{11}$$

- Williams %R
  Williams %R is a technical indicator based on momentum that also determines conditions for stock prices that are overbought and oversold, as follows:

$$R = \frac{\max\left(H\right) - C}{\max\left(H\right) - \min\left(L\right)} \times -100 \tag{12}$$

- Stochastic Oscillator
  Stochastic Oscillator follows the speed or momentum of the price. As a rule, the momentum changes before the price. It measures the level of the closing price over a period of time relative to the low-high range. This indicator is calculated as follows:

$$SO = K - D \tag{13}$$

  where $D = SMA\left(K,3\right)$ and

$$K = \frac{C\left(t\right) - \min\left(L\left(t\right)\right)}{\max\left(H\left(t\right)\right) - \min\left(L\left(t\right)\right)}.$$

- Price Rate of Change (PROC)
  PROC is a technical indicator that shows the rate of price change over a period of time, as follows:

$$PROC\left(t\right) = \frac{C\left(t\right) - C\left(t-12\right)}{C\left(t-12\right)} \tag{14}$$

- Average Directional Index (ADX)
  ADX is a technical analysis indicator used to assess the strength of the trend, with the following formulation:

$$ADX\left(t\right) = \frac{ADX\left(t-1\right)\times 13 + DX\left(t\right)}{14} \tag{15}$$

  where

$$DX\left(t\right) = \frac{\left(PDI\left(t\right) - MDI\left(t\right)\right)\times 100}{PDI\left(t\right) + MDI\left(t\right)}$$

$$PDI\left(t\right) = \max\left(\left(H\left(t\right) - H\left(t-1\right)\right), 0\right)$$

$$MDI\left(t\right) = \max\left(\left(L\left(t\right) - L\left(t-1\right)\right), 0\right)$$

- Bollinger Bands
  Bollinger bands are a type of statistical chart that characterizes the price and volatility of a financial instrument or commodity over time. Bollinger Bands are made up of three lines: a simple moving average as well as upper and lower bands, which are formulated as follows:

$$\begin{aligned} &BB = SMA\left(C,20\right) \\ &Upper\ BB = BB + D\sqrt{\frac{\sum_{t=a+1}^{a+20}\left(C(t)-BB\right)^2}{20}} \\ &Lower\ BB = BB - D\sqrt{\frac{\sum_{t=a+1}^{a+20}\left(C(t)-BB\right)^2}{20}} \end{aligned} \tag{16}$$

- Logarithmic Return
  The logarithmic return calculates the rate of return on an investment, as follows:

$$\text{Log Return} = \log\left(\frac{C\left(t\right)}{C\left(t-1\right)}\right) \tag{17}$$

B. Data Preprocessing

Generally, stock market data shows a random walk behavior. In a random walks time series, the mean and variance changes over time, i. e. they are non-stationary. The analysis on stationary time series data leads to desirable results. To change our data into a stationary time series, we follow the time-differentiating approach which considers the difference between subsequent time measures. To show the difference between the stationary and nonstationary datasets numerically,

TABLE I
COMPARISON OF NON-STATIONARY AND STATIONARY TIME SERIES

	Mean(1)	Mean(2)	Variance(1)	Variance(2)
Non-Stationary	1184.62	1942.33	39053.20	330413.86
Stationary	-0.10	0.78	230.30	302.71

we split our dataset into two equal groups and then, calculate mean and variance for each group. As Table I demonstrates, non-stationary time series have a notable change in mean and variance over time.

After producing a stationary dataset, we denoise the data utilizing the wavelet transformation. In addition, by applying z-value method on the dataset, the outliers are detected and then removed. First, by using (18), we transform the distribution of the data into the normal distribution where mean is 0 and standard deviation is 1.

$$Z_i = \frac{X_i - \mu}{\sigma} \tag{18}$$

where Z_i is the z-score value of the dataset. The part of data that satisfies the z-score value of $Z_i > 3$ and $Z_i < -3$ are identified as outliers.

At the next step, we reduce the level of noise in the dataset to get a clear picture of the main index trend. To do so, the wavelet transformation is applied on the dataset. The details of wavelet transformation are discussed in [19]. Time series data after noise removal is shown in Fig. 4.

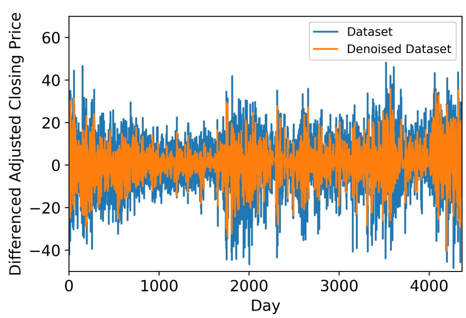

Fig. 4. The raw and reconstructed differenced adjusted closing price of S&P 500 index.

At the last stage, we use the min-max normalization method to normalize the data.

IV. EXPERIMENTS

Our purpose in AE-LSTM is to use prediction for prediction. We use hold-out method for train and test data splitting. 80% of stock data selected for training and the remaining 20% for testing. As an optimization algorithm, we use adaptive moment estimation (Adam) [20] to train our model.

For measuring the accuracy of our model, mean absolute error (MAE), root mean square error (RMSE), and average return (AR) are utilized as the performance criteria, which are formulated as follows:

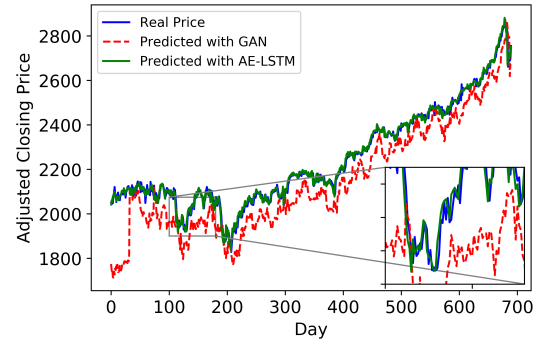

Fig. 5. Adjusted closing price prediction by the proposed AE-LSTM.

TABLE II
THE EVALUATION OF PERFORMANCE CRITERIA

Method	MAE	RMSE	AR
AE-LSTM	0.0054	0.0073	4.34×10^{-5}
GAN	0.2923	0.3553	-9.36×10^{-5}

$$MAE = \frac{1}{N}\sum_{k=1}^{N} |\hat{y}_k - y_k| \tag{19}$$

$$RMSE = \sqrt{\frac{1}{N}\sum_{k=1}^{N} (\hat{y}_k - y_k)^2} \tag{20}$$

$$AR = \frac{1}{N-1}\sum_{k=1}^{N-1} (\hat{y}_{k+1} - y_k),\ if\ \ \hat{y}_{k+1} > \hat{y}_k \tag{21}$$

To demonstrate the efficiency of the proposed method, we have compared the results of the AE-LSTM with the results of GAN as a generative model. The prediction results of the AE-LSTM and GAN are shown in Fig. 5. It can be seen from Fig. 5 that the proposed AE-LSTM approach predicts the adjusted closing price accurately. Also, according to Table II, AE-LSTM has lower MAE and RMSE values than GAN. Therefore, the prediction with the proposed method is almost accurate.

We also generate a buy-sell signal to inform the traders based on the deep learning predictions. To generate this signal, we follow a pair of rules defined as follows:

- If $P(t+1) - SMA(C,5) > \varepsilon_1$ then buy.
- If $SMA(C,5) - P(t+1) > \varepsilon_2$ then sell.

where $P(t+1)$ is the predicted adjusted close price for time $t+1$, $SMA(C,5)$ is the five-day moving average at time t, and ε_1 and ε_2 are the threshold parameters that are determined by the designer.

Fig. 6 illustrates the adjusted closing price along with the trading operations generated according to the above rules with $\varepsilon_1 = \varepsilon_2 = 30$.

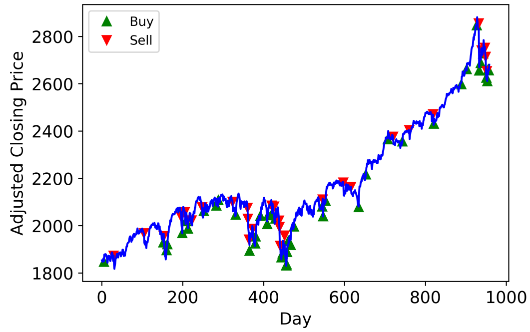

Fig. 6. Generating buy-sell signals according to the predictions and moving average indicator.

V. Conclusion

In this paper, the stock market closing price prediction-by-prediction based on AE-LSTM is proposed. Additional features are created and applied to the dataset, including technical indicators and oscillators. The z-score method efficiently removes the outliers in the dataset. The wavelet transformation has also proved to be a viable approach to reduce market noise. Simulation experiments are carried out on the S&P 500 stock index. According to the results, AE-LSTM is able to predict the daily price almost more accurate than GAN.

References

[1] S. Chen and H. He, "Stock prediction using convolutional neural network," *IOP Conference Series: Materials Science and Engineering*, vol. 435, no. 1, pp. 1–9, 2018.

[2] X. Ding, Y. Zhang, T. Liu, and J. Duan, "Deep learning for event-driven stock prediction," in *Proceedings of the 24th International Joint Conference on Artificial Intelligence*, Buenos Aires, Argentina, pp. 2327–2333, 2015.

[3] R. Singh and S. Srivastava, "Stock prediction using deep learning," *Multimedia Tools and Applications*, vol. 76, no. 18, pp. 18 569–18 584, 2017.

[4] W. Bao, J. Yue, and Y. Rao, "A deep learning framework for financial time series using stacked autoencoders and long-short term memory," *PloS one*, vol. 12, no. 7, p. e0180944, 2017.

[5] M. A. Hossain, R. Karim, R. Thulasiram, N. D. Bruce, and Y. Wang, "Hybrid deep learning model for stock price prediction," in *2018 IEEE Symposium Series on Computational Intelligence (SSCI)*. IEEE, 2018, pp. 1837–1844.

[6] B. Yang, Z.-J. Gong, and W. Yang, "Stock market index prediction using deep neural network ensemble," in *2017 36th Chinese Control Conference (CCC)*. IEEE, 2017, pp. 3882–3887.

[7] T. Gao and Y. Chai, "Improving stock closing price prediction using recurrent neural network and technical indicators," *Neural Computation*, vol. 30, no. 10, pp. 2833–2854, 2018.

[8] Y. Hu, X. Sun, X. Nie, Y. Li, and L. Liu, "An enhanced LSTM for trend following of time series," *IEEE Access*, vol. 7, pp. 34 020–34 030, 2019.

[9] Z. Y. Xu, L. N. Tang, and C. P. Tian, "Prediction of stock trend based on deep belief networks," *Applied Mechanics and Materials*, vol. 644, pp. 5538–5541, 2014.

[10] F. Shen, J. Chao, and J. Zhao, "Forecasting exchange rate using deep belief networks and conjugate gradient method," *Neurocomputing*, vol. 167, pp. 243–253, 2015.

[11] Y. Deng, F. Bao, Y. Kong, Z. Ren, and Q. Dai, "Deep direct reinforcement learning for financial signal representation and trading," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 28, no. 3, pp. 653–664, 2016.

[12] K. Zhang, G. Zhong, J. Dong, S. Wang, and Y. Wang, "Stock market prediction based on generative adversarial network," *Procedia computer science*, vol. 147, pp. 400–406, 2019.

[13] R. T. F. Nazário, J. L. e Silva, V. A. Sobreiro, and H. Kimura, "A literature review of technical analysis on stock markets," *The Quarterly Review of Economics and Finance*, vol. 66, pp. 115–126, 2017.

[14] X. Yan and L. Zheng, "Fundamental analysis and the cross-section of stock returns: A data-mining approach," *The Review of Financial Studies*, vol. 30, no. 4, pp. 1382–1423, 2017.

[15] M. Velay and F. Daniel, "Stock chart pattern recognition with deep learning," *arXiv preprint arXiv:1808.00418*, 2018.

[16] R. Dash and P. K. Dash, "A hybrid stock trading framework integrating technical analysis with machine learning techniques," *The Journal of Finance and Data Science*, vol. 2, no. 1, pp. 42–57, 2016.

[17] M. U. Gudelek, S. A. Boluk, and A. M. Ozbayoglu, "A deep learning based stock trading model with 2-d cnn trend detection," in *2017 IEEE Symposium Series on Computational Intelligence (SSCI)*. IEEE, 2017, pp. 1–8.

[18] Y. Bengio, P. Lamblin, D. Popovici, and H. Larochelle, "Greedy layer-wise training of deep networks," in *Advances in neural information processing systems*, 2007, pp. 153–160.

[19] O. Renaud, J.-L. Starck, and F. Murtagh, "Wavelet-based combined signal filtering and prediction," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 35, no. 6, pp. 1241–1251, 2005.

[20] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proceedings of the International Conference on Learning Representations (ICLR)*, San Diego, CA, USA, pp. 1–15, 2015.