A Financial Deep Learning Framework: Predicting the Values of Financial Time Series With ARIMA and LSTM

Zhenjun Li, Shenzhen Institute of Technology, China*
https://orcid.org/0000-0003-4176-8879
Yinping Liao, Shenzhen Institute of Technology, China
Bo Hu, Shenzhen Yihuo Technology Co., Ltd., China
Liangyu Ni, Shenzhen Telecom Co., Ltd., China
Yunting Lu, Shenzhen Institute of Information Technology, China

ABSTRACT

Prediction of stock price movement is regarded as a challenging task of financial time series prediction. Due to the complexity and massive financial market data, the research of deep learning approaches for predicting the future price is very difficult. This study attempted to develop a novel framework, named 13f-LSTM, where the autoregressive integrated moving average (ARIMA), for the first time, is one of the technical features, using Fourier transforms for trend analysis and long-short term memory (LSTM), including its variants, to forecast the future’s closing prices. Thirteen historical and technical features of stock were selected as inputs of the proposed 13f-LSTM model. Three typical stock market indices in the real world and their corresponding closing prices in 30 trading days are chosen to examine the performance and predictive accuracy of it. The experimental results show that the 13f-LSTM model outperforms other proposed models in both profitability performance and predictive accuracy.

KEYWORDS

ARIMA, Deep Learning, Finance, Fourier Transforms, LSTM, Prediction, Stock Indices, Time Series

INTRODUCTION

The application of time series prediction in financial fields is generally considered one of the most complicated issues in time series analysis due to the financial data noise and vast factors. Time series forecasting is a research program that has some useful applications in numerous other research fields, especially in the financial fields. Forecasting by time series can help people make wise decisions to reduce the risks of investments. In a time series, time is usually an important variable used for making decisions and predictions. When the authors use time series to predict the trend of the future
financial market, for example, the stock indices, the authors need to introduce some detailed historical data over some time to train a model. Researchers usually use historical data to predict various future events, such as the forecast of stock prices and changes in product sales.

In this study, the authors focus mainly on the multi-feature stock selection and the deep learning models in the financial market’s trend prediction. Specifically, like most of the applications of classic time series models, the authors require strict assumptions regarding the distributions and stationarity of time series. Meanwhile, all units of the input vectors are independent of each other. The authors select 13 typical stock technique features, which include the result of autoregressive integrated moving average (ARIMA). After being denoised by Fourier transforms, they will be input into LSTM for future predictions. The model in the researcher’s study is named 13f-LSTM. The flowchart of the 13f-LSTM model for the financial time series is shown in Figure 1. Then the authors evaluate and compare the predictive accuracy of both 12 features LSTM, called 12f-LSTM in this paper (without ARIMA) and 13f-LSTM for a long-term stock time series prediction. Moreover, the authors will compare 13f-BLSTM and 13-LSTM in model performance and predictive accuracy.

RELATED WORK

The organization of this paper is as follows. In the next section, the authors discuss the existing works related to the research and the advantages of the proposed approach is over the existing works. In the following section, the authors proposed a hybrid model, named 13f-LSTM, with introductions to Fourier transform, ARIMA, LSTM, and its variants. The next sections present the inputs and data resources of 13f-LSTM, followed by the design and analysis of our experiment and its results, and finally, the authors present conclusions from the study.

In the last decade, several approaches, including statistical models and machine learning models, have been proposed to predict financial time series. Most of the traditional statistical models assume that the time series under study is generated from a linear process (Kumar & Murugan, 2013). However, financial time series are essentially dynamic, chaotic, nonlinear, complicated, and highly noisy (Si & Yin, 2013). Machine learning models, such as the support vector regression (SVR; Wikipedia, n.d.a) and especially deep learning models, i.e., artificial neural networks (ANNs; Guo et al., 2014), have been applied successfully in modeling, optimizing, and predicting the financial time SERIES (Lee, 2009). Most of them can capture linear or nonlinear relationships between key features with no prior knowledge about the input data (Atsalakis et al., 2014).

Throughout the traditional statistical techniques, for the quantitative stock market, multi-factor or multi-feature stock selection strategy is the most widely used stock selection model in quantitative
stock investment (Zhang et al., 2018). The basic principle is based on mathematical and statistical methods, testing the validity of a series of factors related to stock prices, combining multiple valid features to establish a quantitative model to model stocks, and selecting the best performance according to the corresponding principles for an excess return of the stock portfolio (Peng et al., 2022).

Traditional statistical techniques applied in time series analysis are usually based on the assumption and relationship of linearity among generally distributed variables (Wang et al., 2011). These models for sequential time data, involving some different features to fit a proposed time series model, include autoregressive moving average (ARMA), autoregressive (AR), and autoregressive integrated moving average (ARIMA; Li et al., 2015; Zhang et al., 2016) and its variation, generalized autoregressive conditional heteroskedasticity (GARCH). However, some specific combined parameters of financial time series will make these data more difficult to predict compared with other proposed time series prediction techniques. In any case, after decades of testing, using traditional statistical techniques are proved to be an effective methodology for predicting financial markets.

In machine learning, especially in deep learning models, a lot of research has been performed on predicting time series. A deep recurrent neural network (RNN) is one of the methods employed by many prediction methods; it is a type of neural network used to process sequence data by remembering the preceding data inputs while calculating network weights based on current data. Then long short-term memory (LSTM), proposed as a variation of the deep RNN, was developed to optimize and improve the RNN network performance by considering previous model states and calculating long-term relationships as well. The traditional LSTM was developed to remember the long-term state by adding one more input and output based on the RNNs model, which made it deal with longer input series than RNNs (Hochreiter & Schmidhuber, 1997; Gers et al., 2000). Another variation of RNNs is the bidirectional LSTM (BLSTM). It is an extension format of the original LSTM that can improve network performance by training through the preceding and succeeding input classification sequences. In other words, the BLSTM model trains two instead of one LSTM network on the input sequence. The architecture of a BLSTM model is shown in Figure 4.

However, in the literature, a popular developmental trend in deep learning methods is usually applied in time series prediction. A novel improvement over the traditional statistical techniques is the new ones can successfully deal with dynamic, chaotic, nonlinear, and complex actual real data by extracting distributed factors that show the relevant information (Hinton & Salakhutdinov, 2006) and achieving a better performance than before (Bengio et al., 2013). Considering the continuity and complexity of financial time series analysis, the deep learning models are regarded as some of the most charming methods in financial time series prediction (Cavalcante et al., 2016). One of the main challenges in applying machine learning methods to forecast the financial time series data is to model high-dimensional, noisy, and complex real-world time series via a hierarchical and represented model of the data (Wei et al., 2017). This paper focuses on this topic and proposes a novel model based on the deep learning model.

According to Wei et al. (2017), a primary activity in determining a computational model able to forecast the trend of financial markets is the selection of inputs and outputs of the deep learning model. The provided methods differ according to the types and number of features selected in modeling financial markets (Xia et al., 2015). How to combine the features selections and optimization with the deep learning models to make higher forecasting precision in the financial market is a relatively unexplored field.

One of the novel methods used to predict the future is TEI@I methodology, which was proposed by Professor Wang. The TEI@I methodology combines traditional statistical and artificial intelligence. It can be applied to analyze the future trend of complicated systems that are unstable, emergent, uncertain, or nonlinear. It is based on the integration of text mining, economic econometrics, and intelligence. In this methodology, econometrical models are used to deal with the linear factors of time series, while nonlinear factors of the time series are modeled by using an artificial intelligence model, such as ANN. The TEI@I methodology has achieved amazing results in forecasting the trends
of exchange rates, import and export volume, food production, GDP, FDI, CPI, money supply and demand, housing prices, and commodity prices. However, there are few linear components in the time series of stock markets, which need a new methodology that combines traditional statistical and artificial intelligence for it. In our work, the authors proposed a new deep learning framework that combines traditional statistical and artificial intelligence to achieve the one-step-ahead financial time series prediction.

**METHODOLOGY**

To activate the one-step-ahead financial time series prediction, the authors propose a deep learning framework named 13f-LSTM for stock prices prediction using multi-features stock selections and long-short term memory (LSTM). Before feeding them the data, which should be preprocessed, enhanced in multiple steps:

1. Features selection and optimization by more technical indicators. Autoregressive integrated moving average, one of the most popular methods to predict future values of time series data, will be added to the features for the predictions.
2. Data preprocessing by using the fast Fourier transform, which is developed to decompose the financial markets time series for eliminating noise.

Figure 1 shows the flow chart of the 13f-LSTM model framework. The described knowledge of each block is further presented as follows.

**Fourier Transform**

The Fourier transform decomposes a function of time or a signal into its constituent frequencies. The Fourier transform and its variant, fast Fourier transform (FFT), is applied to deal with the unstable and complex financial time series data for predicting future trends (“Fourier transform,” 2019). In this study, the authors will use the Fast Fourier transforms, which is used in computational finance mostly, for predicting the prices of financial derivatives on a time series a basis (Nicolson, 2018).

The Fourier transform of a function $f$ is traditionally denoted as $\hat{f}$ by adding a circumflex to define the symbol of the function. There are some common formulas to define the Fourier transforms of an integrable function $f: R \rightarrow C$. One of them is:

$$\hat{f}(\xi) = \int_{-\infty}^{\infty} f(x) e^{-2\pi i \xi x} dx \quad (1)$$

for any real number $\xi$.

One of the reasons for the negative sign in the exponent is that it is common in financial or electrical engineering to represent by $f(x) = e^{2\pi i 0 x}$ a signal with zero initial phase and frequency $\xi_0$. The negative sign formula causes the product $e^{2\pi i 0 x} e^{-2\pi i \xi x}$ to be 1 (frequency zero) when $\xi = \xi_0$ causing the integral to diverge. The result is a Dirac delta function at $\xi = \xi_0$, which is the only frequency component of the sinusoidal signal $e^{2\pi i 0 x}$.

When $x$, the independent variable, represents time and the transform feature $\xi$ represents frequency. Under a suitable condition, $f$ is defined by $\hat{f}$ by the inverse transform:

$$f(x) = \int_{-\infty}^{\infty} \hat{f}(\xi) e^{-2\pi i \xi x} d\xi \quad (2)$$
for any real number $x$.

When the statement that $f$ can be reconstructed from $\hat{f}$, the authors call it the Fourier inversion theorem, which was first introduced in Fourier’s analytical theory of heat, although what would be considered proof by modern standards was not given until much later. Both the functions $f$ and $\hat{f}$ usually are referred to as a Fourier integral pair or Fourier transform pair. The fast Fourier transform is applied for extracting global and local trends in the financial market indices and also to denoise them. The authors can use fast Fourier transforms to extract long- and short-term trends by using the transforms with different components (“Fourier transform,” n.d.).

The authors will use the Fast Fourier transforms on the close prices of Ping An Bank Co.Ltd. (000001.SZ) as an example of data optimization. The period is from 18/04/2016 to 16/04/2018. As you can see in Figure 2, the more components from the fast Fourier transforms the authors used, the closer the approximation function is to the real stock closing prices. (When setting the 50 components transforms in the function, it is almost identical to the original function. As seen in Figure 2, the purple and the red lines almost overlap.) When the financial time series is very rough, to denoise it, the data curve will be smoother by using fast Fourier transforms. In this study, the authors used FFTs with nine component transforms.

**Autoregressive Integrated Moving Average**

In econometrics and statistics, and especially in time series analysis, an autoregressive integrated moving average (ARIMA) model is fitted to time series data better to both understand the data and predict future values in the series (Wikipedia, n.d.b). The definition of ARIMA is as follows (“Autoregressive integrated moving average,” n.d.).

Given a time series of data $X_t$ where $t$ is an integer index and the $X_t$ are real numbers, an ARMA $(p, q)$ model is given by:

$$X_t - \alpha_1 X_{t-1} - \ldots - \alpha_p X_{t-p} = \varepsilon_t + \theta_1 \varepsilon_{t-1} + \ldots + \theta_q \varepsilon_{t-q}$$

or equivalently by:
\[
\left(1 + \sum_{i=1}^{q} \theta_i L^i \right) \varepsilon_i = \left(1 - \sum_{i=1}^{p'} \alpha_i L^i \right)x_i
\]

where \(L\) is the lag operator, the \(\alpha_i\) are the factors of the autoregressive part, \(\theta_i\) are the parameters of the moving average part of the model, and the \(\varepsilon_i\) are error terms of the model. The error terms \(\varepsilon_i\) are usually assumed to be identically distributed, independent features sampled from a normal distribution with zero means.

Now assume that the polynomial has a unit root (a factor \((1 - L)\)) of multiplicity \(d\). Then it can be rewritten as:

\[
\left(1 - \sum_{i=1}^{p'-d} \phi_i L^i \right)(1 - L)^d = \left(1 - \sum_{i=1}^{p'} \alpha_i L^i \right)
\]

An ARIMA \((p, d, q)\) process expresses this polynomial factorization property with \(p'd = p\) and is given by:

\[
\left(1 + \sum_{i=1}^{q} \theta_i L^i \right) \varepsilon_i = \left(1 - \sum_{i=1}^{p'} \phi_i L^i \right)(1 - L)^d X_t
\]

and that can be known as a special case of an ARMA \((p + d, q)\) function, which has the autoregressive polynomial with \(d\) unit roots. (For this reason, no ARIMA model with \(d > 0\) is wide sense stationary.)

The above statements can be described as follows:

\[
\delta + \left(1 + \sum_{i=1}^{q} \theta_i L^i \right) \varepsilon_i = \left(1 - \sum_{i=1}^{p'} \phi_i L^i \right)(1 - L)^d X_t
\]

ARIMA model is one of the techniques to predict time series data. Its performance of it on a real stock (000001.SZ) is illustrated in Figure 3. The predicted period is from 18/04/2016 to 16/10/2016. The model gives an amazing approximation of the stock closing prices. However, in this study, ARIMA will not provide the final prediction. The authors used the price predicted by ARIMA as an input factor feeding into the deep learning model, as the authors mentioned above, to analyze as many patterns and features of stock indices as possible. In addition, the model also can help to denoise the financial data.

Technical Indicators

Selecting input features is a useful and representative task to forecast effectively. Many factors can be irrelevant or redundant for the modeling process. To determine which feature sets are better fitted to a financial time series forecasting model, some preprocessing jobs were investigated in the use of extraction mechanisms or automatic factor selection. The main advantage of features extraction or factors selection is dimensionality reduction, which consequently helps to reduce the computational complexity and efforts of the deep learning models used. At the same time, it will reduce the risk of overfitting, which is a modeling error that occurs when a function is too closely fit a limited set of data points. The elimination of irrelevant and minor factors leads to better predictive performances.
The technical indicators of the stock are relative to the fundamental analysis, focusing on the analysis of the operating management status of each company, the general economic situation, the industrial environments, and the other features to measure the financial indices. The technical analysis is to investigate the market behavior response through the records of charts. The main content of the technical indicators on which it is based is to calculate from historical data such as price index, index futures, stock prices, and trading volume.

There are more than 100 technical indicators in the financial market, excluding derivative indicators of various combinations. In this work, 13 features will be selected to train the predictive model. They are described in Table 1.

Table 1. The description of the input features

<table>
<thead>
<tr>
<th>No.</th>
<th>Name</th>
<th>Descriptions</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Open</td>
<td>nominal daily open price</td>
</tr>
<tr>
<td>2</td>
<td>Close</td>
<td>nominal daily close price</td>
</tr>
<tr>
<td>3</td>
<td>High</td>
<td>nominal daily highest price</td>
</tr>
<tr>
<td>4</td>
<td>Low</td>
<td>nominal daily lowest price</td>
</tr>
<tr>
<td>5</td>
<td>Volumn</td>
<td>daily trading volume</td>
</tr>
<tr>
<td>6</td>
<td>EMA12/26</td>
<td>12/26 day exponential moving average</td>
</tr>
<tr>
<td>7</td>
<td>MA7/MA21</td>
<td>7/21 day moving average</td>
</tr>
<tr>
<td>8</td>
<td>Momentum</td>
<td>pinpoint the end of a decline or advance</td>
</tr>
<tr>
<td>9</td>
<td>MACD</td>
<td>Moving average convergence divergence</td>
</tr>
<tr>
<td>10</td>
<td>BOLL</td>
<td>Bollinger band: provides a relative definition of high and low</td>
</tr>
<tr>
<td>11</td>
<td>ARIMA</td>
<td>AutoRegressive integrated moving average</td>
</tr>
</tbody>
</table>
Long-Short Term Memory

Long short-term memory is one of the several variations of recurrent neural networks (RNNs) architecture (How et al., 2014). In this section, the models of LSTM and its variants to forecast the stock index price will be introduced as follows. The authors start with the LSTM model and then proceed to its variant, the bidirectional LSTM (BLSTM). As mentioned before, an LSTM network is a type of RNN. The RNN is one of the methods employed by many prediction techniques; it is a type of neural network used to process sequence data by remembering the preceding data inputs while calculating network weights based on current data (Hinton et al., 2012; Dahl et al., 2012). RNNs have a deep framework in the temporal dimension, which makes it useful in time series analysis and modeling (Palangi, Deng, et al., 2016; Palangi, Ward, & Deng, 2016) because they keep track of all previous data values and can capture disciplines and patterns developing during a period. Due to its nature, the gradient disappears; that is, the weight changes during training become so small that they will not change, so that the network cannot converge to the minimum loss (the opposite problem can also be observed). When the gradient becomes too large, this is called gradient exploding, but the solution is very simple: Clip gradients if they exceed some constant number, i.e., gradient clipping.

The LSTM model is well-suited to processing, classifying, and predicting the stock prices based on time series data since there can be lags of unexpected duration among several important events in a time series prediction. The traditional LSTM model was developed to solve the vanishing gradient and exploding problems that can be encountered when training the original RNN network. Relative insensitivity for gapping length is an advantage of the LSTM model over the RNN model, which was hidden Markov models and other machine learning models in numerous applications. The LSTM transition equations are described as follows (Graves et al., 2005):

\[
\begin{align*}
i_t &= \sigma \left( W_i x_t + U_i h_{t-1} + V_i c_{t-1} \right) & (7) \\
f_t &= \sigma \left( W_f x_t + U_f h_{t-1} + V_f c_{t-1} \right) & (8) \\
o_t &= \sigma \left( W_o x_t + U_o h_{t-1} + V_o c_t \right) & (9) \\
c_t &= f_t \odot c_{t-1} + i_t \odot \tilde{c}_t & (11) \\
h_t &= o_t \odot \tanh (c_t) & (12)
\end{align*}
\]

where \( i_t \) is the input gate and \( o_t \) is the output gate. The memory cell, hidden state, and forget gate are denoted by \( c_t \), \( h_t \), and \( f_t \), respectively, as shown in Figure 4.
One of the several variations of LSTM is the bidirectional LSTM (BLSTM), which as proposed by Schuster and Paliwal (1997) to train the model using input time-series data sequences in the future and past. There are two connected layers, which are used to deal with the input data. In each layer, by using reversed time step direction, to perform the operations. The results can then be combined using different types of merge methods. Similarly, the bidirectional LSTM model uses two layers for data training. One of the layers performs the training following the same direction of the data sequence and the other one processes its operations in the reverse direction of the data sequence. The bidirectional LSTM has been found more effective than traditional LSTM in some applications, such as phoneme classification (Graves et al., 2005).

![Figure 4. The architecture of a BLSTM model](image)

**RESEARCH DATA**

In this section, the authors describe the research data resource and the selection of input features of the model.

**Data Resource**

All of our research data were downloaded from Yahoo finance (https://finance.yahoo.com/). The authors coded the proposed stock predictive methodology in Python using Google Colaboratory, which is a free Jupyter notebook environment that requires no setup and runs entirely in the cloud. The three stock indices for predictions the authors choose are CSI 300, Nikkei 225, and S&P 500 index. Sample data from different markets can help to deal with a problem in which market environment, policy, and state may potentially influence the validity of the deep learning network. The S&P 500 index, trading in the New York Stock Exchange, is generally considered the most mature financial market in the world. On the contrary, financial markets in mainland China are often known as new ones whose policies and environments are still far from being fully completed. So, the authors choose the CSI 300 index as a sample market for developing markets. In addition to the markets described above, the Nikkei 225 index in Tokyo represents a market between the developing and developed ones. These three stock indices, as the research data, provide a natural setting to test the comprehensiveness and robustness of model performances based on different market conditions. The authors downloaded the three stock indices’ data sets for the period from 01/01/2018 to 31/12/2021 from Yahoo finance. The total number of cases was 975 trading days. This period is just during some special international events, such as the trade war between China and the USA and the COVID-19 pandemic, which were considered to be black swan events to global stock markets. Selecting experimental data in this period can effectively avoid model training issues caused by abnormal data.
Input Features

The authors selected thirteen features from several historical and technical indicators as input features. Table 1 describes their details. The features include the historical trading data of each index like opening price, high price, low price, closing price, trading volume, and the technical factors of each index like EMA 12/26 and MACD. ARIMA is also one of the technical indicators.

By focusing on daily changes, preprocessed data is more stable than before. Firstly, using Fourier transforms to denoise the data. Then investigated the difference between every consecutive pair of days. Lastly, the data would be normalized between 0 and 1 using min-max normalization or other methods. In this study, the authors segmented the data set into three subsets: a training set, a validation set, and a testing set. The first 80% duration of the whole data (765 trading days) was allocated for training, about 17% (180 trading days) was allocated for validation, to tune hyperparameters for getting a more advanced model weight, and the remaining 3% (30 trading days) was allocated for testing. The same proportion was used to split the time scale into three stock indices’ data. For measuring the proposed model’s predictive accuracy and performance compared with the actual data, the authors used four performance indicators to measure the results of our experiments: root mean square error (RMSE), mean absolute error (MAE), coefficient of determination, which is denoted as R2, and the accuracy of the error (AOE). In better prediction, RMSE and MAE should be closer to 0 thus R2, AOE should be closer to 1. Four measures are described mathematically as follows:

\[
MAE = \frac{1}{n} \sum_{t=0}^{n-1} |y_t - \hat{y}_t| \\
RMSE = \sqrt{\frac{1}{n} \sum_{t=0}^{n-1} (y_t - \hat{\gamma}_t)^2} \\
R^2 = 1 - \frac{\sum_{t=0}^{n-1} (y_t - \hat{\gamma}_t)^2}{\sum_{t=0}^{n-1} (y_t - \bar{y})^2} \\
AOE = 1 - \sqrt{\frac{1}{n} \sum_{t=0}^{n-1} \frac{(y_t - \hat{\gamma}_t)^2}{y_t^2}}
\]

where \(y_t\) and \(\hat{y}_t\) represent the real and predicted stock indices values, respectively, at step \(t\) (\(0 \leq t < n\)) and \(\bar{y} = \sum_{t=0}^{n-1} y_t / n\). Another question in our deep learning model, which has a few more parameters, is likely to produce overfitting. The first method the authors used to prevent it was to increase the amount of training data. the authors allocated 80% duration of the whole data (765 trading days) to train the LSTM network. The other approach to avoiding overfitting was defining the validation set (about 17% of the whole dataset), which is set to simulate the test set. The reason for evaluating the model on the validation set is that developing a deep learning model usually requires us to adjust its configuration of it. The authors make this adjustment through the model’s performance.
on the validation set. In essence, this adjustment is also a learning process: finding the best model configuration in a certain feature space.

**EXPERIMENT**

After finishing the data preprocessing, the next step is the experimental part. Here, the authors present experimental results and evaluate the performance of the proposed models.

**Experiment Design**

The first purpose of the experiment is to compare models’ performance after inputting new factor ARIMA in different markets of stocks. The second one is to predict performance between LSTM and BLSTM in the same conditions. Other than that, a comparative analysis will show the differences in the same model before and after using Fourier transforms.

In the first experiment, for the CSI 300, Nikkei 225 and SP500 index, LSTM is fed by 12 features and 13 features (include ARIMA) each. The prediction procedure consists of three parts. During the training procedure, the authors feed the LSTM model with a 12 × 30 and 13 × 30 array of indices data within a sliding window to calculate its weights. The authors use a window of size 30 to predict 30 days. The following period of six months (a half year) is applied for validating. Then the authors will predict the future stock indices with the test data. At last, MAE, RMSE, R2, and AOE will be calculated to analyze the performance of the different models.

In the second experiment, the data, 13 features of three markets indices, would be trained in LSTM and BLSTM to predict the future trends in one month. The performance of the two models would be compared with the results of AOE. The third part of the experiment would show how Fourier transforms work in the model by eliminating noise and creating approximations of the real stock movement. The same data, before and after Fourier transforms, would be fed into the LSTM.

**RESULTS**

However, the results show that the performance of 13f-LSTM is not improved significantly, for some indicators are the same as 12f-LSTM, because only including one vector ARIMA has limited changes to the model. Anyway, the results show that considering the correct factors in training can improve the accuracy of the deep learning model, which is also what the authors will focus on in the next step: including more factors to make the financial time series prediction more accurate.

The results of the first experiment are shown in Figure 5 and Table 2. For every stock index, the authors show a month predicted results from LSTM with 12 and 13 features and the corresponding real data in a graph. It is necessary to note that the prices of stocks have been normalized between 0 and 1. According to Figure 5, the authors can find that the LSTM model with 12 features has larger distances variations to the real data than with the 13f-LSTM model. Table 2 shows the records of the models’ performance measures in forecasting three stocks. It can be seen from the table that the 13f-LSTM model shows better performance than the 12f-LSTM model in predicting all three

<table>
<thead>
<tr>
<th>Markets</th>
<th>MAE</th>
<th>MASE</th>
<th>R2</th>
<th>AOE</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>12f</td>
<td>13f</td>
<td>12f</td>
<td>13f</td>
</tr>
<tr>
<td>Nikkei 225</td>
<td>0.017</td>
<td>0.015</td>
<td>0.049</td>
<td>0.044</td>
</tr>
<tr>
<td>S&amp;P 500</td>
<td>0.016</td>
<td>0.015</td>
<td>0.047</td>
<td>0.045</td>
</tr>
<tr>
<td>CSI 300</td>
<td>0.001</td>
<td>0.001</td>
<td>0.005</td>
<td>0.004</td>
</tr>
</tbody>
</table>
For example, for predicting the Nikkei 225 index, the values of MAE and MASE of 13f-LSTM were 0.015 and 0.044, respectively, which is less than those of the 12f-model. Besides this, the R2 value is 0.994, which is better. The same results happened in the predictions of SP500 and CSI 300 markets.

The result of the second experiment is very simple and clear, which is shown in Table 3. The models of 13f-LSTM and 13f-BLSTM have a similar performance in predictive accuracy. For example, in Nikkei 225, the accuracy of 13f-LSTM is better than 13f-BLSTM with 0.936 vs. 0.934, but it just shows the opposite results in S&P 500 prediction.

The BLSTM model combines the forward LSTM and backward LSTM. LSTMs have been used successfully in a multitude of real-world problems, from classical time-series issues as described here to text auto-correction, anomaly detection, fraud detection, and self-driving car technologies. There are currently some limitations to using the LSTM model. As of now, however, BLSTMs provide significant advancements on more classical statistical time-series approaches in being able to model the relationships non-linearly and being able to process data with multiple dimensions in a nonlinear fashion.

Figure 6 illustrates the role of the Fourier transform in time series prediction. It shows that after Fourier transforms, the curve is smoother and more accurate, which helps to denoise the data preprocessing. It also works in the predictive process.

The Fourier transform is applied for data denoising in some data preprocessing since it can handle the non-stationary financial time series data. The key property of Fourier transforms is that it can analyze the frequency components of financial time series with time simultaneously. As seen in Figure 6, after Fourier transforms, the prediction shows a better performance, which increased by over 30%. The wavelet transform is also a method for data denoising, which is also introduced in our models in the next step.

<table>
<thead>
<tr>
<th>Table 3. The predictive accuracy of 13f-LSTM and 13f-BLSTM</th>
</tr>
</thead>
<tbody>
<tr>
<td>Market</td>
</tr>
<tr>
<td>Nikkei 225</td>
</tr>
<tr>
<td>SP500</td>
</tr>
<tr>
<td>CSI 300</td>
</tr>
</tbody>
</table>
CONCLUSION

In this paper, the authors propose a novel and effective forecasting framework to predict the trend of three typical stock indices in different financial markets. The procedure to build this forecasting framework is as follows: First, to select features and optimize from all kinds of stock technical indicators. Autoregressive integrated moving average (ARIMA), one of the most popular techniques for predicting future values of time series data, adds to the feature pool for the predictions. Second, the authors used Fourier transforms to preprocess the data, which helped in decomposing the stock price time series for eliminating noise. Third, a long short-term memory (LSTM) model was used to generate the straightforward output by training effectively. The input features include the daily OHLC features, technical indicators, and ARIMA. The main contribution of this work is that it is the first attempt to combine traditional statistical techniques (ARIMA) with deep learning (LSTM) in financial time-series predictions. In addition, the framework, 13f-LSTM model, is proposed with several modules to denoise multi-feature stock selection instead of factors selection and financial time series fitting.

The authors test the profitability and predictive accuracy of our proposed 13f-LSTM model compared with the 12f-LSTM model. The results provide evidence that it shows better performance in both profitability and predictive accuracy regardless of which data set is for the experiment. Although the proposed 13f-LSTM model has satisfactory predictive performance, it still has some insufficiencies. For example, the reasons why the model performance improves after ARIMA was added can’t be explained in theory. An advanced problem is which features should or should not introduce in the model will improve the model’s accuracy. In addition, deep learning methods are time-consuming. More RNN variants, such as GRU, stacked LSTM, should be tested instead of LSTM and BLSTM. Wavelet transforms, as a method for data denoising, may help to improve our models. All of these could be enhanced by future studies.

ACKNOWLEDGMENT

This work was supported by Shenzhen Institute of Technology Project [2111010]; Guangdong Philosophy and Social Sciences Planning Project [GD21CYJ21]; Guangdong Province Key Laboratory of Popular High Performance Computers [2017B030314073]; Guangdong Educational Science Planning Project [2019GXJ237]; and The key planning project of education and scientific research of Shenzhen Institute of Education[ZD2021003]. Dr. Bo Hu is the corresponding author of this paper.

FUNDING AGENCY

Open Access Funding for this article has been covered by the authors of this manuscript.
REFERENCES


Zhenjun Li, male, doctoral candidate, with the title of senior engineer of electronic technology, is now the director of the teaching and Research Office of communication network application specialty of Shenzhen Technician College. His research interests include deep learning blockchain, data mining, etc.

Liao, Y. female, received the M.S. (2006) from Huazhong University of Science and Technology, lecturer at the School of Information and Communications, Shenzhen Institute of Technology. Research Interests: Computer Vision, Machine Learning, Deep Learning. PhD, graduated from Wuhan University

Liangyu Ni was born in Guangdong, China, 1998. He has been graduated from the Computer Science and Technology, Shenzhen University. He received a scholarship to College of Computer Science & Software Engineering, Shenzhen University in 2017. He won a Certificate of Achievement in 2019 Interdisciplinary Contest in Modeling, and was designated as Honorable Mention. His current research interests include web and deep machine learning.

Lu Yunting, female, doctor and associate professor, has successively presided over one project of Guangdong Provincial Natural Science Foundation, two projects of Shenzhen educational science planning, and four school teaching and scientific research projects. As a main researcher, she has participated in the research work of many scientific research projects, including the National Natural Science Foundation of China and Guangdong Provincial Natural Science Foundation, Won the second prize of Guangdong science and Technology Progress Award and one of Shenzhen Science and technology progress award. He has published more than ten relevant academic papers in various journals at all levels and international conferences.