TEST Engineering & Management

March - April 2020
ISSN: 0193-4120 Page No. 4315 - 4321

Research on Stock Price Prediction Technology using RNN and Y Text Miner

Sunghyuck Hong[*1], Jungsoo Han[2]
[*1,2]Associate Professor, Division of Information and Communication Technology, Baekseok University, Cheonan, Republic of Korea
shong@bu.ac.kr[*1], jshan@bu.ac.kr[2]

Article Info
Volume 83
Page Number: 4315 - 4321
Publication Issue:
March - April 2020

Article History
Article Received*: 24 July 2019*
Revised*: 12 September 2019*
Accepted*: 15 February 2020*
Publication*: 26 March 2020*

Abstract

The stock price prediction system has been developed and, it is being used. Most of them use the time series inference to analyze past stock price analysis data and the recent artificial intelligence based deep learning to learn the pattern of stock price fluctuations and the future. However, there are many factors that affect stock prices, and it is literally impossible to forecast the future with historical data, and the stock market will not be able to repeat past trading patterns, and it will respond to new situations at every moment including international relations, specific events, and celebrity comments. Because of the impact, even with deep learning algorithms, it is nearly impossible to predict exact the price of stocks that change over time. In addition, the 2013 Nobel Prize winner in Economics Eugene Fama mentioned that it can make profits from the market with a temporary strategy, but if other market participants grasp the pattern, and the strategy is not effective. Therefore, it is impossible to continue to win the market. Realistically, Eugene Fama mentioned that it could be able to make a profit if one wants to make a profit, and then it responds to a changing market. Therefore, based on historical data, finding the closest situation to when the stock price has risen will be able to predict the stock price continuously. This research collects all the possible news texts from the Internet by using YTextMiner and analyses time series data from past stock market using RNN(Recurrent Neural Network). This research results can effect on all kinds of prediction system such as earthquake prediction, weather forecast, crime prediction, and election.

Keywords: *Classification, Machine Learning, Apache Spark , Hadoop.*

1. Introduction

The stock price prediction system has been developed and is being used, but most of them use the time series inference to analyze past stock price analysis data and the recent artificial intelligence based deep learning to learn the pattern of stock price fluctuations and the future. We are using a system to predict. However, there are many factors that affect stock prices, and it is literally impossible to predict the future with historical data, and the stock market will not be able to repeat past trading patterns, and will respond to new situations at every moment, including international relations, specific events and celebrity comments. Because of the impact, even with deep learning algorithms, it is simply impossible to know the price of stocks that change over time. In addition, the 2013 Nobel Prize winner in Economics Eugene Fama created an efficient market hypothesis that, while a temporary strategy could make money in the market, it is impossible for other market participants to grasp the pattern so that if the strategy is not effective, it will not be able to continue to win the market. However, realistically Eugene Fama may be able to make a profit if one can make a profit and respond to a changing market. Therefore, based on historical data, finding the closest situation to when the stock price has risen will be able to predict the stock price continuously. In this project, we analyze the past stock price analysis using RNN (Recurrent Neural Network) and analyze the influence on the stock price by searching the real-time stock news by using inference through time series and YTextMiner (Text Miner) [1]. The innovation of this project is the stock price prediction technology using circulating neural networks (RNN) and news mining, which can be applied to securities firms and investment institutes by developing a specialized stock price analysis model for each stock by developing a program that predicts the short-term stock price. The forecasting program is expected to generate high profits from business models due to its diverse application.

Additional information such as KOSPI index, KOSDAQ index, exchange rate information, Bollinger Bands, trading value, supplementary indicators, stock news, etc. that affect stocks are analyzed and predicted through RNN.

LSTM (Long Short Term Memory) networks are a type of RNN, and LSTM is continuously multiplied by values between [-1, 1] by Chain Rule, which is a long-term dependency problem. It can solve the problem that the parameters are not updated and disappear. It is the algorithm proposed by Hochreiter [3]. The existing RNN simply calculated St = tanh (Uxt + WSt) when calculating the hidden state (St) in the hidden layer. However, there are four calculation processes in LSTM. As shown in the hidden layer of LSTM below, there are four Neural Network Layers. Figure 1 shows the four Neural Network Layers.

The *C* value, which flows like a conveyor belt on Figure 1, is the cell state, and the LSTM has three gates to protect and control this cell state: forget, input, and output gates to prevent vanishing gradients and allow the gradient to flow effectively.

- Forget gate f_t is literally a gate to forget past input. Because the output range of the sigmoid function is 0 to 1, if value is 0, the data of the previous state is forgotten and if it is 1, the data of the previous state is completely stored.

- Input gate is a gate for remembering current input. This value is 0 ~ 1 because it is a sigmoid, but the $\hat{c}_t$ of hadamard product is -1 ~ 1 because it is hyperbolic tangent result. Therefore, the result may be negative.

- The output gate o_t is the gate for the final result h_t, and the final product of the LSTM is the product of the hadamard product of the hyperbolic tangent of the cell state.

Keras is a Python library that makes deep learning easy. Keras is a high-level interface that makes it easy for developers to implement deep learning. Keras makes it easy to implement deep learning, it can develop an artificial based stock price prediction system [4].

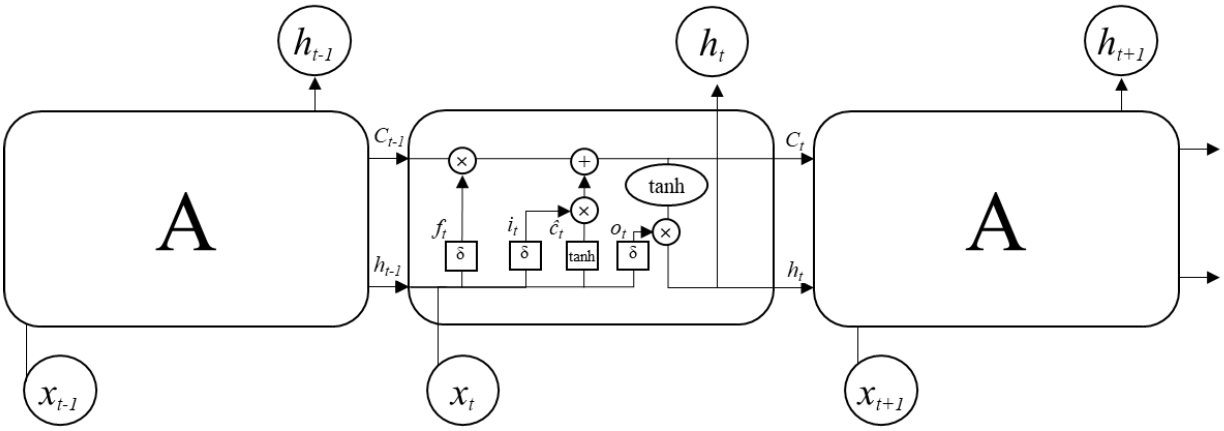

Figure 1. Structure of LSTM.

2. Related Work

The development of prediction systems using artificial intelligence has past data because it can be applied in a variety of fields, such as weather forecasts, climate change, earthquake occurrence, typhoon movement routes, virus movement routes, and car movements. Research for predicting the future is actively progressing. Among them, stock price prediction is known in the field where technology development is most actively performed due to the large market size. A lot of effort has been put into stock price forecasting because it is necessary to determine buying and selling timing through continuous monitoring. However, applying artificial intelligence deep learning techniques to analyze past data If the news affecting the stock price is automated by text mining and the number and sales are made via a trading bot, the selling and buying can proceed at the optimal time without unnecessary monitoring. The need to develop artificial intelligence-based automated trading bots is necessary to minimize losses through emotional investment through technical analysis and to buy and sell shares more efficiently.

In the United States, most systems trade through automated trading programs. There is a clear standard through technical analysis, not an automatic trading program and simply unexplained line of trading machines, there is a clear standard through this algorithm, trading and raising to professional investor level Marketing can be carried out, selection of market price, number of products, sale number according to the approach Split sale bulk sale according to the trading period, no emotional loss and next paragraph, mechanical marketing of artificial intelligence If you get rid of the most poisoned sentiment in the stock market and seize the cost of a loss due to soaring and falling when time investment is not possible, then take the stock with two rabbits: expertise and time. It will be able to do predict.

This artificial intelligence-based automatic stock forecasting research is an IT program developed to open a new era to a healthy stock culture through a standard investment and unscrupulous stock investment in the past indiscriminate stock investment. And can provide tools to become successful investors in the right stock culture.

3. Proposed Stock Prediction System

Many programs that analyze artificial intelligence-based time-series data to predict stocks are under development, but stock news that affects stocks is still text-mining and weighted stock analysis is still being performed. We expect this product to be highly competitive.

3.1. Test Environments

The testing code is based on Python 3. We used Samsung Electronics past data from Yahoo finance. The data we use is contained in the 005930.csv file and consists of the closing price of the Yahoo finance from January 2016 to January 2020.

3.2. Test Software

We used Jupyter notebook[5], Tensorflow[6], and Keras. Tensorflow is an end-to-end open source platform for machine learning. It has a comprehensive, flexible ecosystem of tools, libraries and community resources that lets researchers push the state-of-the-art in ML and developers easily build and deploy ML powered applications. The Jupyter Notebook is a widely open web application tool that it can be used for creating and sharing documents which contain text, visualizations, equations, and live code. In addition, Jupyter Notebook is maintained by the people at Project Jupyter. Jupyter Notebooks are a spin-off project from the IPython[7] project, which used to have an IPython Notebook project itself. The name, Jupyter, comes from the core supported programming languages that it supports: Julia, Python, and R. Jupyter ships with the IPython kernel, which allows you to write your programs in Python, but there are currently over 100 other kernels that can be used as well[8-11].

3.3. Data set and Testing Software Environments

Currently, Python 3.7 is available. However, Python 3.7 doesn't fully support Keras library. Thus, we used Python 3.6 instead the current version. There is a minor modification to use this code, if you use Python 3.7 in Table 1[12].

Table 1. Python 2.0 source code for data import

```
import datatime
From pandas as pd
from keras.models import Sequential
from keras.layers.core import Dropout, Activation,
Dense
from keras.layers.recurrent import LSTM

#Step 1: import for past stock past data
data=pd.read_csv('005930.KS.csv')
data.head()

#Step 2: Calculate average price
high_prices = data['High'].values
low_prices = data['Low'].values
average_prices = (high_prices + low_prices) / 2
```

Table 2 is Samsung Electronics stock dataset from Feb. 2016 to Feb. 2020. If No.1 is standard, the stock prices are transferred as less than 1 or larger than -1 which is normalized process. The incremental in Table 2 is Middle(No.1) / Middle(No.2) -1. Pandas in Table 1 is enable to load data as *.csv file from Yahoo finance. If you run data.head() in Jupyter Notebook[1], it will show the result as Table 2. We computed average price (average_prices) between high and low, and then calculate incremental. After calculation of average price, we created windows in Table 3[12].

Table 2. Samsung Electronic past stock data set

No	Date	High	Low	Middle	Incremental
1	2019-02-08	45,700	44,650	45,175	0
2	2019-02-11	45,000	44,250	44,625	0.0123
3	2019-02-12	46,250	44,650	45,450	- 0.0182
4	2019-02-13	46,700	46,000	46,350	- 0.0194
5	2019-02-14	47,500	46,150	46,825	- 0.0101
…	…	…	…	…	…

[1] Jupyter Notebook is an open-source web application which is Python web based editor, and it can allow you to create and share documents which contain live code, equations, visualizations and narrative text.

Table 3. Create Windows

```
seq_len = 30
sequence_length = seq_len + 1
result = []
for index in range(len(middle_prices) - sequence_length):
    result.append(middle_prices[index: index +
sequence_length])
```

Seq_len in Table 3 means set of training set, so it starts with No.1 thru No. 30 in Table 2. We used 30 training set that is 30 days stock trading data set. After training, it predicts the next day price so that stock investors can decide whether they sell or buy stocks. In order to predict price well, we normalized data set in Table 4. We divided train and test data set with 9:1 ratio (row = int(round(result.shape[0] * 0.9)). 0.9 in Table 4 means that training set is 90% of data set, and 0.1 means that test set is 10% of data set. It depends on which stock you will predict for future[12,13].

Table 4. Normalizing data process

```
# Normalized data set
normalized_data = []
for window in result:
    normalized_window = [((float(p) / float(window[0]))
- 1) for p in window]
…
result = np.array(normalized_data)

# Divide train and test data set
row = int(round(result.shape[0] * 0.9))
…
x_test = np.reshape(x_test, (x_test.shape[0],
x_test.shape[1], 1))
y_test = result[row:, -1]
…
```

Now, it is time to build a model. Because of time serial data, we built a model as Sequential() function from Keras library. Table 5 shows which model we used for Samsung Electronics. Sequential model is very common in most stock prediction case[12].

Table 5. Sequential model

```
model = Sequential()
model.add(LSTM(30, return_sequences=True,
input_shape=(30, 1)))
…
```

After building a sequential model, training get started as Table 6. We used fit method in model. Batch_size in Table 6 is how many data we can train in a row. Epochs is repeat count and unit of data set. We used 90% of training set and 10% of test set which is confirmed for validation. 0.9 in Table 4 is divided training set and test set. Each set is different, and it has never been overlapped[12].

Table 6. Training process

```
model.fit(x_train, y_train,
    validation_data=(x_test, y_test),
    Batch_size=5, Epochs=5)
…
```

Table 7. Model Results

```
Layer (type)     Output Shape               Param #
=====================================================
lstm_1 (LSTM)   (None, 30, 30)              3840
_____________________________________________________
lstm_2 (LSTM)   (None, 64)                  24320
_____________________________________________________
dense_1(Dense)  (None, 1)                   65
=====================================================
Total params: 28,225
Trainable params: 28,225
Non-trainable params: 0
```

Table 7 shows the model execution result. LSTM is library in Keras and it trained 30 sets which means 30 days stock past data from Yahoo finance. (None, 1) in Table 7 means 1 day prediction. Therefore, we trained 30 days and received future 1 day for Samsung stock price.

4. Conclusion

This study relates to a real-time securities information prediction program, and more particularly, to a real-time securities information prediction program of a new concept that enables users to easily determine the securities to purchase. Recently, as the Internet is activated, securities information providing sites that provide securities information using Internet communication are increasing. The stock information providing site is able to transmit the stock information displayed on the market board of the current stock exchange in real time through the Internet homepage, and the user can check the stock information in real time without visiting the stock exchange floor. It can be used as a source of information. However, such a stock information providing site is difficult to accurately determine which stocks to buy and at what point by providing information about the stock price to the user without any analysis or recommendation. Therefore, a method has been developed in which a stock broker recommends several stocks so that a user can buy and sell stocks of recommended stocks. However, such a system has a problem in that it is difficult to judge a stock based on stock trends in real time. Therefore, a new concept of real-time securities information forecasting program is needed to predict the stock price according to the stock price in advance and display the information about 15 minutes ahead of actual trading point to help users' trade stocks. This research has been devised to solve the above problems, and an object of the present invention is to help users to trade stocks and increase the user's investment profit by providing the user with a prediction of the stock price according to the stock price trend in advance. It is to provide a new concept of real-time securities information prediction program. The real-time stock information prediction program as described above may be implemented by a stock price inference method through time series analysis, and the function may be implemented through information such as stock price trends, politics, stock news, and economic status. In conclusion, stocks have to be invested and are being turned into speculative sentiments. Therefore, this study contributes not only to investing emotionally but also to establishing a sound investment culture through technical analysis.

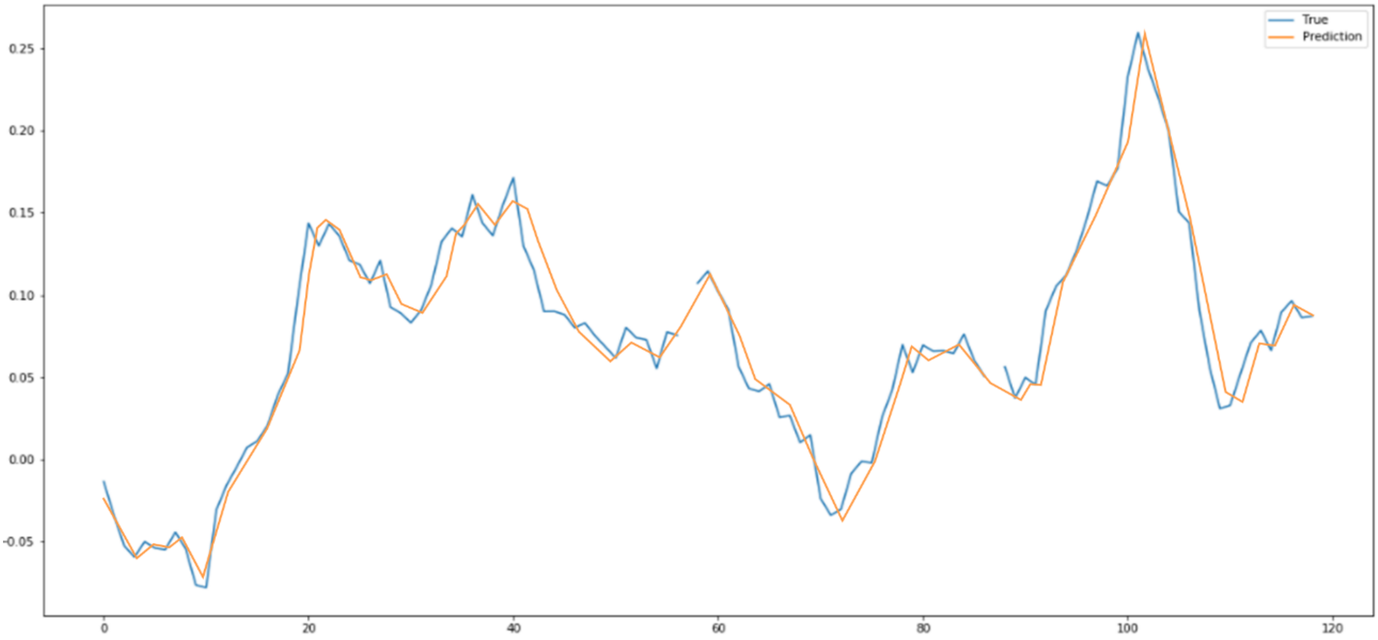

Figure 2. Prediction Graph

Acknowledgment

This research is supported by 2020 Baekseok University Research fund.

References

[1] Optimizing Deep Learning RNN Topologies on Intel Architecture. Supercomputing Frontiers and Innovations. 2019;6(3).

[2] Warrick P, Nabhan Homsi M. Ensembling convolutional and long short-term memory networks for electrocardiogram arrhythmia detection. Physiological Measurement. 2018;39(11):114002.

[3] Hochreiter R. Applied Mathematical Programming and Modelling 2016. ITM Web of Conferences. 2017;14:00001.

[4] Bilashenko S, Shapovalova N, Rybalchenko O. Image recognition by convolutional neural networks with using Keras library. Mining Journal of Kryvyi Rih National University. 2018;(103):147-153.

[5] Jupyter Notebook: An Introduction [updated 2019 June 15; cited 2020 Feb 5]. Available from: https://realpython.com/jupyter-notebook-introduction/

[6] Bandi R, Amudhavel J. Object Recognition Using Keras with Backend Tensor Flow. International Journal of Engineering & Technology. 2018;7(3.6):229.

[7] Python Algorithms: mastering basic algorithms in the Python language. Choice Reviews Online. 2011;48(10):48-5731-48-5731.

[8] Detlefsen K. Monte Carlo Option Pricing in Tensorflow. SSRN Electronic Journal. 2018.

[9] Zhao B, Li X, Lu X, Wang Z. A CNN–RNN architecture for multi-label weather recognition. Neurocomputing. 2018;322:47-57.

[10] G S, Kp S, R V. Automated detection of diabetes using CNN and CNN-LSTM network and heart rate signals. Procedia Computer Science. 2018;132:1253-1262.

[11] Reddy B, J.C U. Prediction of Stock Market using Stochastic Neural Networks. International Journal of Innovative Research in Computer Science & Technology. 2019;7(5):128-138.

[12] Taehee Lee. Stock Crypto Price Prediction [Internet]. [updated 2018 Oct 31; cited 2020 Feb 19]. Available from: https://github.com/kairess/stock_crypto_price_prediction

[13] Sunghyuck Hong. Development of a Secure and Intelligent IoT System based on a Consortium Blockchain. International Journal of Emerging Multidisciplinary Research. 2017;1(2):9-14.