

Predicting Stock Prices using LSTM and a Sliding Window Approach

Abstract

With proper knowledge of any companies stock and insight, one can gain large profit sitting at home. Stock market of a company is a time series data and stock trend/price prediction is one of the field where many researchers had gathered interest to predict the stock prices or trend in future using historical data and technical indicator with high accuracy. A good prediction model of a stock's future price will increase trader's profits. In this report, the proposed model uses deep learning model LSTM — Long short term memory to predict the stock trend in a sliding window approach.

Index Terms— Long Short Term Memory, Historical Data, Technical Indicators and Stock price Prediction

1. Introduction

Stock price prediction is a topic that attracts many data researchers and data analytics as a good stock prediction can capitulate notable profit. Stock of a company cannot just be predicted easily Stock market is volatile, haywire, uncertain and non-linear data. Finding patterns in these stocks is a very difficult task. The random change in the stock market is referred as random-walk behavior of stock prices with time. Firstly for prediction of any stock price, we need to analyze the data. For analysis two approaches are used Fundamental and Technical indicators used to forecast stock prices. After analyzing the data various linear and non-linear models are being used to predict the data like ARIMA (auto regressive model) and ANNs. Artificial Neural networks like RNN, CNN, and LSTM are commonly used model in stock price prediction [1]. ANN was inspired from function of human brain and implementing a complex network of neurons. In [2], they proposed a fusion model implementing HMM, ANN and GA for predicting stock price. These Models are widely used in areas like Image Processing, Natural Language Processing, Time Series Analysis, etc. Over-fitting and under-fitting of data is a big problem while using ANN model for stock price prediction [3]. ANNs are

very useful for short term forecasting. While non-linear model are better choice to predict stocks, many factual researchers had shown that non-linear models might not outperform linear models every time [4] [5] [6].

In this model, I have implemented LSTM model. LSTM was proposed back in 1997 by Felix Gers and his adviser Jürgen Schmidhuber and Fred Cummins who introduced the forget gate to deal with the vanishing gradient problem as stated in [7]. LSTM is one of the most important model because of the introduction of forget gate and memory cell. In this model, the information flows through a mechanism known as cell states. Due to these memory cells now, LSTM selects and remember or forget things according to its importance. Therefore, LSTM can learn and identify patterns of data dynamically with time and produces huge prediction accuracy. LSTM was first used for time series forecasting long back in 2002 by FA GersD, EckJ and Schmidhuber in [8].

2. Methodology

For Prediction of Stock Market, we need to deal with huge historical data that is highly nonlinear. To deal with this high non-linearity we need to find hidden pattern in our data and analyze them for prediction of future prices. Yet pattern identi-

fication given a nonlinear data is a trivial task and therefore there is a need of a dynamic model that could analyze our data and find all the hidden patterns. ANNs are very useful and capable of finding all the hidden patterns and exploiting the data to predict the future prices through self-learning process. These Neural networks are very efficient to predict the stock future prices and therefore are widely used. To predict a financial time series data Using Neural networks was introduced in [9]. In this report, I have used Long Short Term Memory as a prediction model to predict the stock price of Netflix using Historical data of past 17 years from <https://finance.yahoo.com/>.

In this report I have divided the prediction approach into subtopics and the subtopics are as follows:

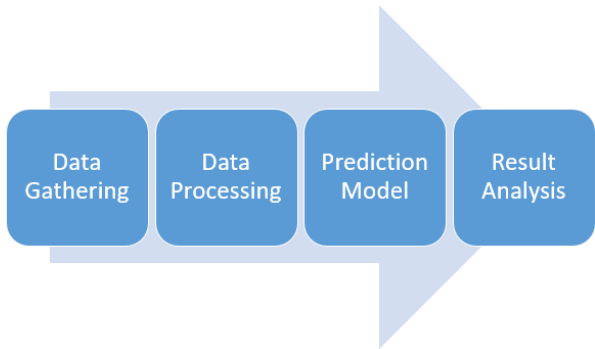


Figure 1: Sequence Flow of this Report

2.1. Data Gathering

17 years of data of Netflix from March 2002 to March 2019 is used in this prediction model. All the data has been collected from <https://finance.yahoo.com/> and downloaded under the historical data section. This Historical data is used to predict the future stock prices.

2.2. Data Processing

2.2.1. Data Extraction

The historical data gathered was raw unprocessed data with high volatility. Prediction using this raw data is not a good option so first we need to process this data. Therefore, I have calculated technical indicators. Technical Indicators are the detailed study of past Market action for

the purpose of forecasting future prices. It helps in forecasting the price direction and the current trend.

Technical Indicators
Simple Moving Average - SMA
Exponential Moving Average - EMA
Triple Exponential Moving Average - TEMA
Kaufman's Adaptive Moving Average - KAMA
Moving Average Convergence/Divergence - MACD
Bollinger Bands
%B
Relative Strength Index - RSI
Average True Range - ATR
Chandelier Exit - CE
Chande Momentum Oscillator - CMO
Force Index - FI
Elder-ray
Stochastic %k
Stochastic %D
Williams %R
Accumulation Distribution Oscillation - ADO
Commodity Channel Index - CCI

2.2.2. Correlation

After extracting all these features we cannot use all these features in our model the as not all features are relevant some of them are irrelevant and introduces noise in our model. Also, having redundant features confuses our model and therefore increases the computational time. Therefore, we only need to select those features that are related to our stock price, and we could discard other features.

The selection of features was based on the correlation coefficient value of all these features with the original stock's closing prices. The features with the highest correlation value was selected. The correlation techniques used over here where Scatter diagram and Pearson Correlation value.

Scatter Diagram

. Scatter diagram is a graph which represents the

relation between to data set. In this graph the values of the two data set are plotted along the two axes and the pattern of the resulting graph gives a basic idea about the correlation between the two data set.

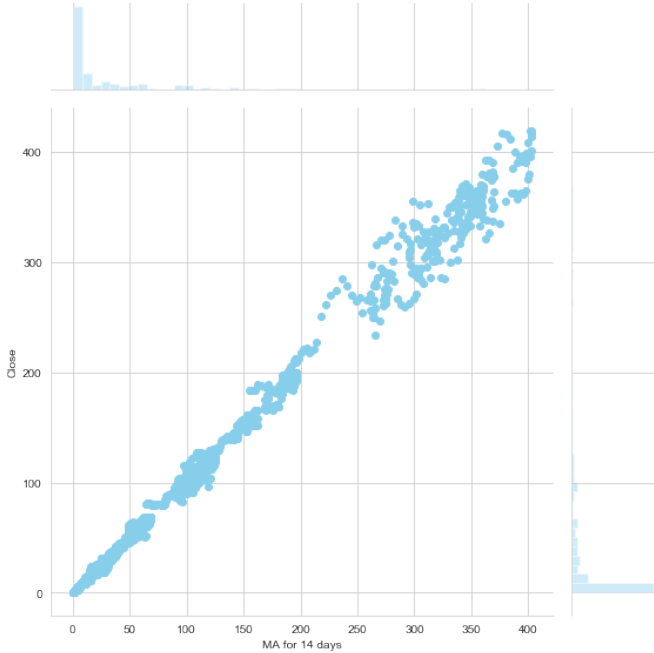


Figure 2: Scatter Diagram for SMA for 14 days vs Close

Pearson Correlation. Pearson Correlation gives a numerical response for finding the relation between different data set. It assigns a number to the extent of relation between two datasets. Its value lies from -1 to 1, 1 representing exactly linear relation between two datasets and 0 representing no relation.

$$r = \frac{\sum((x - \bar{x})(y - \bar{y}))}{\sqrt{\sum(x - \bar{x})^2 \sum(y - \bar{y})^2}} \quad \bar{x} = \text{mean of } x$$

2.2.3. Data Transformation

After getting the best feature, the next thing we do is data transformation. Data transformation is used to normalize the data and make the data stationary which helps in pattern finding. Normalization helps improve convergence of the data. The data was transformed/mapped in the range 0 to 1.

After the data set is transformed into a clean data set, the data set is divided into training and

testing sets so as to evaluate the prediction accuracy of my model. The training set is 95 percent of the total data set and the testing data is the rest of the data.

3. Prediction Model

LSTM is the most powerful RNN architecture.

After normalizing the data this data was fed into the LSTM network for training. The model was trained for 40 epochs and a batch size of 60. Initially the number of epochs were 100 and changed to find a good prediction model. This LSTM model was initialized of an input sequential layer lead by 4 LSTM layers each having neurons lesser than the previous and then finally a dense output layer with Adam optimizer.

```

1 regressor = Sequential()
2 regressor.add(LSTM(units = 60,
   return_sequences = True,
   input_shape = (X_train.shape
     [1], 1)))
3 regressor.add(Dropout(0.2))
4 regressor.add(LSTM(units = 45,
   return_sequences = True))
5 regressor.add(Dropout(0.2))
6 regressor.add(LSTM(units = 30,
   return_sequences = True))
7 regressor.add(Dropout(0.2))
8 regressor.add(LSTM(units = 15))
9 regressor.add(Dropout(0.2))
10 regressor.add(Dense(units = 1))
11 regressor.compile(optimizer = '
   adam', loss = '
     mean_squared_error')
12 regressor.fit(X_train, y_train,
   epochs = 40, batch_size = 60)

```

Code Snippet 1: Code for LSTM Neural Network

4. Experimental Result

After performing continuous simulations for different number of features and epochs, we have observed that by taking MA14 feature with 40 epochs we are able to achieve nearly the best results with training RMSE of

Table 2: Epoch = 40, Feature input = MA14

Input	RMSE	Size
Train	2.5983488911838393	393
Test	6.39236469202595	223

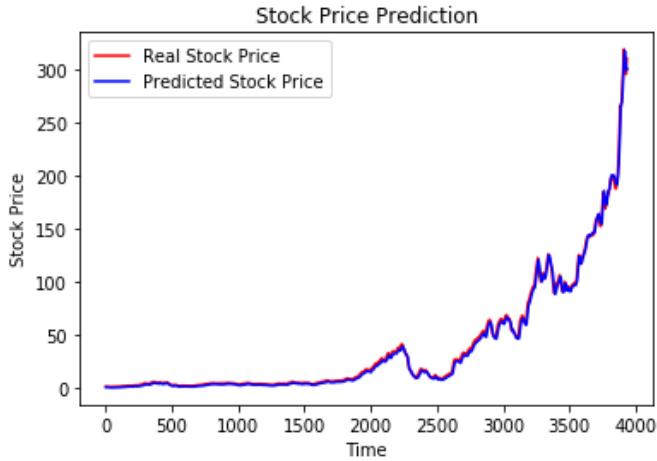


Figure 3: Result of LSTM model for Train Dataset

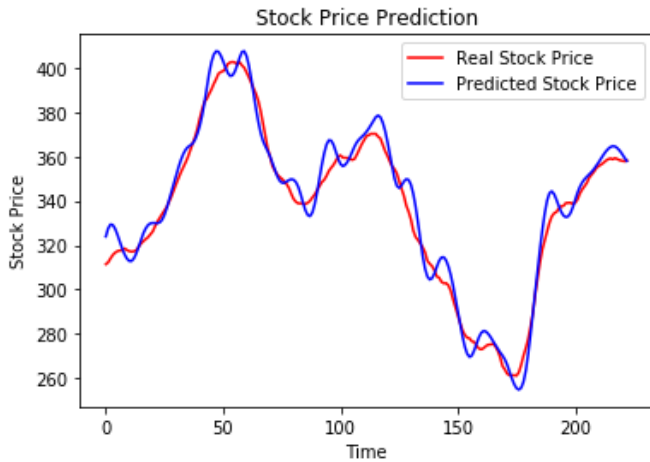


Figure 4: Result of LSTM model for Test Dataset

2.5983488911838393 and testing RMSE of **6.39236469202595**.

5. Conclusion

Forecasting stock prices are very much helpful for investors to earn huge profit. Predicting future price of a given stock to produce an accurate result is encouraging researchers to find some new technique to improve the accuracy. RNNs

like LSTM are very good in processing sequential time series data. LSTM has been proven a very good solution while dealing with sequential data streams. In this work, we have been able to produce significantly good result using a sliding window approach and LSTM model to predict the future price of Netflix stocks.

References

- [1] O. Coupelon, Neural network modeling for stock movement prediction A state of the art, Blaise Pascal University (2007) 83.
- [2] M. R. Hassan, B. Nath, M. Kirley, A fusion model of HMM, ANN and GA for stock market forecasting, Expert systems with Applications 33 (1) (2007) 171–180.
- [3] X. Tao, H. Renmu, W. Peng, X. Dongjie, Input dimension reduction for load forecasting based on support vector machines, in: 2004 IEEE International Conference on Electric Utility Deregulation, Restructuring and Power Technologies. Proceedings, Vol. 2, 2004, pp. 510–514.
- [4] A. A. Adebiyi, A. O. Adewumi, C. K. Ayo, Comparison of ARIMA and artificial neural networks models for stock price prediction, Journal of Applied Mathematics 2014.
- [5] C. K. Lee, Y. Sehwan, J. Jongdae, Neural network model versus SARIMA model in forecasting Korean stock price index (KOSPI), Issues in Information System 8 (2) (2007) 372–378.
- [6] K. J. Lee, A. Y. Chi, S. Yoo, J. J. Jin, Forecasting korean stock price index (kospi) using back propagation neural network model, bayesian chiao's model, and sarima model, Journal of Management Information and Decision Sciences 11 (2).
- [7] F. A. Gers, J. Schmidhuber, F. Cummins, Learning to forget: Continual prediction with LSTM.
- [8] F. A. Gers, D. Eck, J. Schmidhuber, Applying LSTM to time series predictable through time-window approaches (2002).
- [9] H. White, Economic prediction using neural networks: The case of IBM daily stock returns.