Is Sliding Window Approach better for Stock Trend Prediction?

Ankit Thakkar*, Mufaddal Naya

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 price prediction is one of the fields where many researchers had gathered interest to predict the stock prices or trend in future using historical data and technical indicator with high accuracy. Simple Moving Average is very useful in predicting the future price direction and gives a good assumption about the future price. 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, long short term memory (LSTM) to predict the stock trend with two different approaches, with and without using a sliding window and comparing their results.

Index Terms– Deep Neural Network, Long Short Term Memory, Sliding Window Approach, Simple Moving Average, 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 [1]. The stock price of any given company is uncertain. Finding patterns in these stocks is a difficult task. The random change in the stock market is referred as random-walk behavior of stock prices with time by [2]. This statement holds as there are many uncertain factors as the country's progress, any natural disaster or the political status of the country.

To predict the stock of given data we need to find patterns in the stock chart using the raw data or extracting technical indicators. Firstly, for prediction of any stock price, we need to analyze the data. For analysis, two approaches are used: Fundamental and Technical analysis used to forecast stock prices. Fundamental analysis usually deals with the cause of the market, it takes all the macro economic factors such as the company's growth, the climate etc. to predict the trend of the future stocks. Technical analysis create extra data variables using mathematical equations, derived from raw data, and plot it on stock charts to analyze the patterns in the stock price. After analyzing the data various linear and non-linear models are being used to predict the data like ARIMA (auto regressive model) and Artificial Neural Networks (ANNs).

2. Related Work

Deep Neural Networks(DNN) like Recurrent Neural Network (RNN), Convolutional Neural Network (CNN), and LSTM are commonly used model in stock price prediction [3]. ANN was inspired from function of human brain and implementing a complex network of neurons. In [4], they proposed a fusion model implementing Hidden Markov Model (HMM), ANN and Genetic Algorithm (GA) for predicting stock price. These Models are widely used in areas such as Image Processing, Natural Language Processing, Time

*Corresponding author.

Email addresses: ankit.thakkar@nirmauni.ac.in (Ankit Thakkar), 17bit055@nirmauni.ac.in (Mufaddal Naya)

Series Analysis, etc. Over-fitting and under-fitting of data is a big problem while using DNN model for stock price prediction [5]. 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 [6] [7] [8]. Recently in [9], authors tried to compare linear model with nonlinear model and tried to find the accuracy, which shows how the nonlinear model outperform linear model. The linear model in the comparison was ARIMA whereas the nonlinear model were GRU and LSTM in which the LSTM outperform every other model. Also in [10], authors tried to predict the effect of demonetization on stocks of seven Indian companies such as CNX, NIFTY50. ANN were used to predict the future values of these stocks. [11] here they used Deep learning models to predict the stock price movement and analyzed the accuracy of many models such as LSTM, CNN, RNN and many other nonlinear models. Also in [12] compared Support Vector Machine (SVM), Back Propagation and LSTM and analyzed the accuracy.

2.1. LSTM with Sliding Window Approach

In this work we compared two different approaches for time series forecasting using LSTM, with and without sliding window approach. Sliding window approach is predicting F(t+1) considering values from F(t), F(t-1), F(t-2) etc. The similar idea is discussed in [13]. In this work we compared LSTM model with and without Sliding window approach and show how without using any sliding window approach we can get better result compared to model using sliding window approach. We used LSTM to predict the future prices for both the model. LSTM was first proposed in [14] by Felix Gers and his adviser Jürgen Schmidhuber and Fred Cummins who introduced the forget gate to deal with the Vanishing Gradient problem. RNN had a major drawback which was it cannot remember values which had been explored at the beginning of the training set and therefore the training process takes a long time and the accuracy of the model decreases, as discussed in [14]. In LSTM 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. FA GersD, EckJ and Schmidhuber first used LSTM for time series forecasting long back in 2002 in [15]. In [16], authors used LSTM to predict the stocks of China using 30 days sliding window approach. Sliding window approach using LSTM is frequently used as in [17], authors compared LSTM, RNN and CNN using sliding window approach. In [18], authors investigated the effectiveness of LSTM with sliding window approach.

3. 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 identification 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. DNNs 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 [19]. In this report, we 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 work we have divided the prediction approach which are as follows:

3.1. Data Gathering

In this work, 17 years of data of Netflix from March 2002 to March 2019 is used. All the data has been collected from https://finance.yahoo.com/ and downloaded under the historical data section. The content of the downloaded data were:

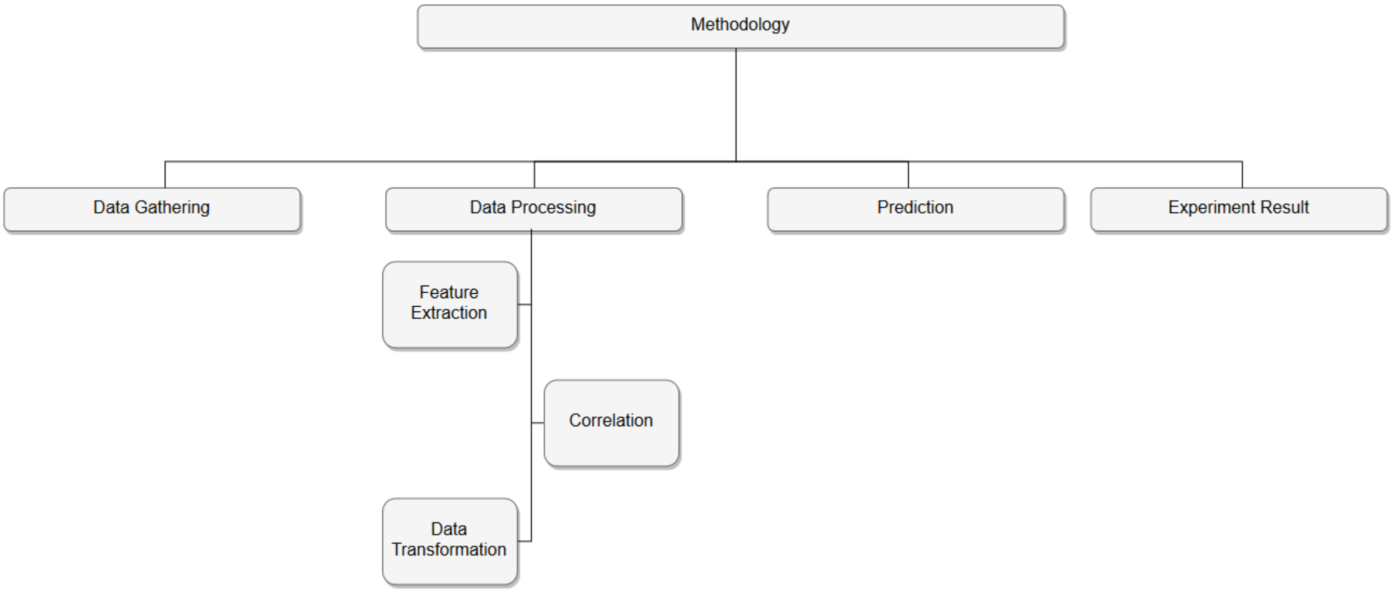

Figure 1: Structure of this paper

low, high, open, close, volume and adj close. This historical data is used to predict the future stock prices using LSTM Architecture.

3.2. Data Processing

Before we train the LSTM model, we need to process the data. Processing is done by extracting some of the features from the stock price and normalizing the data.

3.2.1. Feature Extraction

The historical data gathered was raw unprocessed data with high volatility. Prediction using this raw data is not a good option, because the raw data is volatile and uncertain, so first we need to process this data. Therefore, we 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.

Some of the technical indicators which we studied in this work are listed in Table 1

3.2.2. Correlation

After extracting all these features we cannot use all these features in our model, because not all features are relevant some of them are irrelevant

Table 1: Technical Indicators

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

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. 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.

The scatter diagram for SMA and closing price of the data is shown inFigure 2

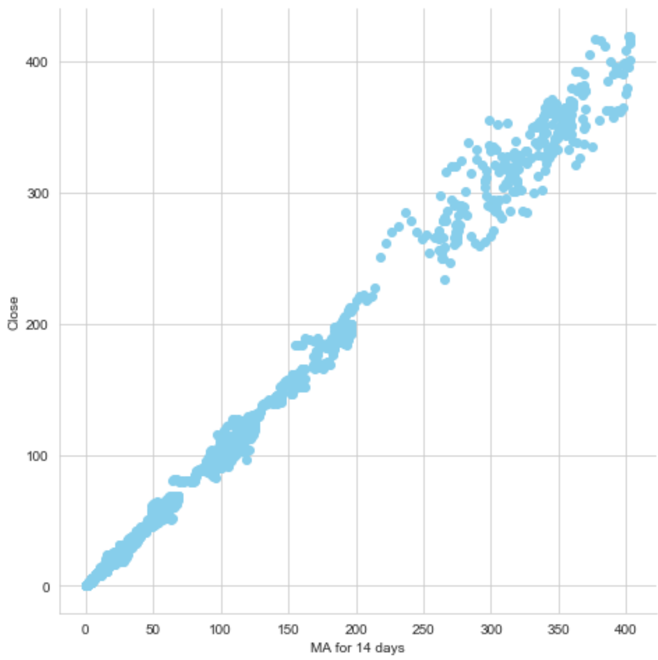

Figure 2: Scatter Diagram for SMA vs Close

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 data sets. Its value lies from -1 to 1, 1 representing exactly linear relation between two data sets and 0 representing no relation.

The formula for calculating Pearson correlation coefficient is Equation (1)

$$r = \frac{\sum((x - \overline{x})(y - \overline{y}))}{\sqrt{\sum(x - \overline{x})^2 \sum(y - \overline{y})^2}} \tag{1}$$

After analysis the Data using Pearson correlation and verifying using scatter diagram Simple moving average for 14 days was chosen as the parameter for the input for stock prediction for Netflix. A SMA for 14 days of closing price of a stock is defined as the rolling average of closing price of the stocks over the last 14 days. A SMA helps in smoothing out the curve which helps in reducing the volatility in the curve. SMA is used for predicting the trend of the direction of price in the future.

3.2.3. 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 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.

4. Prediction Models

In this work, we have implemented LSTM architecture and compared two different approaches with and without sliding window to understand which one performs better. LSTM was introduced to have long termed dependencies and deal with the vanishing gradient problem.

4.1. LSTM with Sliding Window Approach

The model was trained for 40 epochs and a batch size of 60. 60 days sliding window approach was used to predict the future trend. Initially the number of epochs were 100 and changed accordingly to build 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 and loss mean square error.

4.2. LSTM without Sliding Window Approach

The input to this architecture is moving average of previous day and predicting the next day moving average. The Architecture of this LSTM model is same as the previous one with 40 epochs and a batch size of 60. Also, the network consist of one input layer with any sliding window 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 and loss mean square error.

5. Experimental Result

Root Mean Square Error (RMSE) is used to calculate the error for each model. The parameter setting for both the models is given in Table 2. The result analysis table for both the model is presented in Table 3.

Table 2: Parameter setting for both the Models

Parameter	Value
Input	SMA 14
Hidden Layer	4
Optimizer	Adam
Loss Function	mean square error
Epoch	40
Batch Size	60

5.1. LSTM with Sliding window

The result for training and testing data is shown in Figure 3Figure 4 respectively.

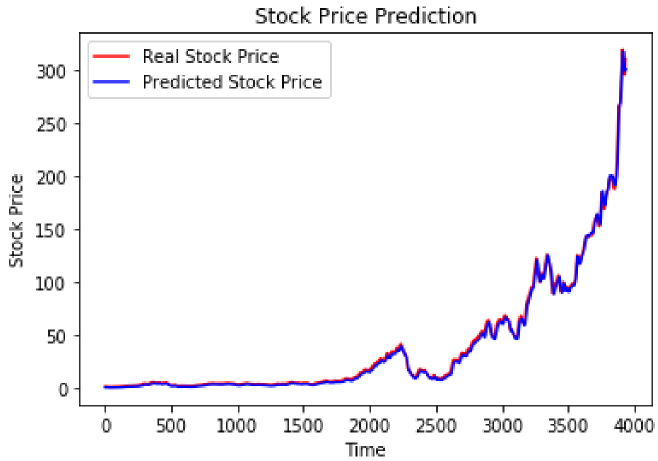

Figure 3: Result of LSTM with sliding window model for Train data set

5.2. LSTM without Sliding window

The result for training and testing data is shown in Figure 5Figure 6 respectively.

6. Conclusions

In this work, we tried to compare two different approaches to understand which performs better. We predicted SMA of stocks of Netflix using sliding window and without using sliding window and compared the result. From the results it was clear that without using a sliding window gives better result compared to with using sliding

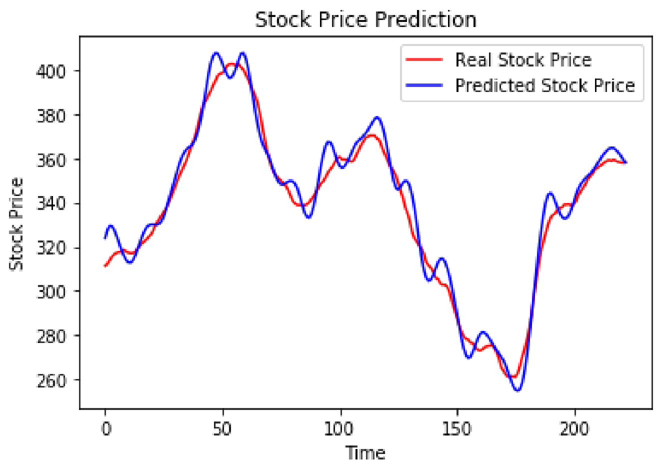

Figure 4: Result of LSTM with sliding window model for Test data set

Table 3: Result Analysis

Approach	Input	RMSE	Size
With sliding window	Train	2.5983	3934
	Test	6.3923	223
Without sliding window	Train	2.8778	3994
	Test	3.5498	211

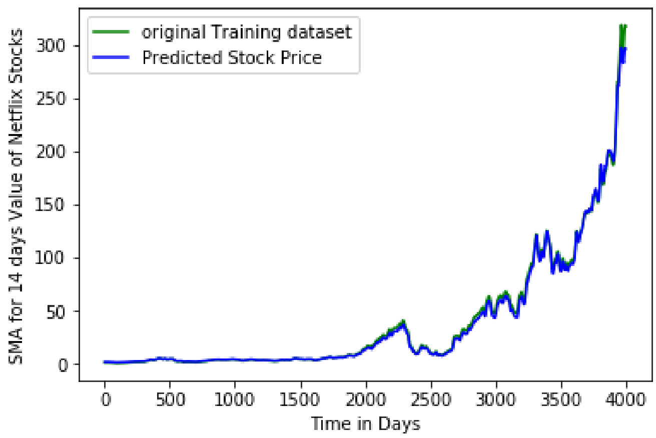

Figure 5: Result of LSTM without sliding window model for Train data set

window. SMA for 14 days was used as an input parameter for both the approaches. We would like to highlight forecasting stock prices or trend are very much helpful for investors to earn huge profit. Different methods are adopted to predict

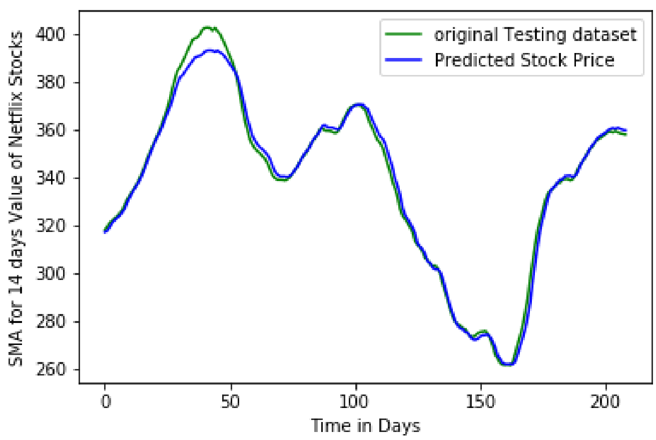

Figure 6: Result of LSTM without sliding window model for Test data set

the future price. Predicting future price or trend 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 at processing sequential time series data. LSTM has been proven a very good solution while dealing with sequential data streams. In this work, we were able to produce significantly good result without using sliding window approach and LSTM architecture to predict the future trend of Netflix stocks by predicting the SMA for the stock.

References

[1] K. C. Brown, W. V. Harlow, S. M. Tinic, Risk aversion, uncertain information, and market efficiency, Journal of Financial Economics 22 (2) (1988) 355–385.

[2] B. G. Malkiel, E. F. Fama, Efficient capital markets: A review of theory and empirical work, {The journal of Finance} 25 (2) (1970) 383–417.

[3] O. Coupelon, Neural network modeling for stock movement prediction A state of the art, {Blaise Pascal University} (2007) 83.

[4] 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.

[5] 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.

[6] 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.

[7] 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.

[8] 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).

[9] X. Zhang, C. Li, Y. Morimoto, A Multi-factor Approach for Stock Price Prediction by using Recurrent Neural Networks, {Bulletin of Networking, Computing, Systems, and Software} 8 (1) (2019) 9–13.

[10] S. Chopra, D. Yadav, A. N. Chopra, Artificial Neural Networks Based Indian Stock Market Price Prediction: Before and After Demonetization, {J Swarm Intel Evol Comput} 8 (174) (2019) 2.

[11] W. Long, Z. Lu, L. Cui, Deep learning-based feature engineering for stock price movement prediction, {Knowledge-Based Systems} 164 (2019) 163–173.

[12] N. Arora, et al., Financial Analysis: Stock Market Prediction Using Deep Learning Algorithms.

[13] E. Zivot, J. Wang, Rolling analysis of time series (2003).

[14] F. A. Gers, J. Schmidhuber, F. Cummins, Learning to forget: Continual prediction with LSTM.

[15] F. A. Gers, D. Eck, J. Schmidhuber, Applying LSTM to time series predictable through time-window approaches (2002).

[16] K. Chen, Y. Zhou, F. Dai, A LSTM-based method for stock returns prediction: A case study of China stock market, in: 2015 IEEE International Conference on Big Data (Big Data), 2015, pp. 2823–2824.

[17] S. Selvin, R. Vinayakumar, E. A. Gopalakrishnan, V. K. Menon, K. P. Soman, Stock price prediction using LSTM, RNN and CNN-sliding window model, in: 2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI), 2017, pp. 1643–1647.

[18] H. Jia, Investigation into the effectiveness of long short term memory networks for stock price prediction, arXiv preprint arXiv:1603.07893.

[19] H. White, Economic prediction using neural networks: The case of IBM daily stock returns.