Comparative Analysis of Time-Series Forecasting Algorithms for Stock Price Prediction

Baleshwarsingh Joosery
Department of Computer Science & Engineering
Manipal Institute of Technology, Manipal Academy of Higher Education
Manipal, India
akashjoosery@gmail.com

G Deepa
Department of Computer Science & Engineering
Manipal Institute of Technology, Manipal Academy of Higher Education
Manipal, India
deepa.ganesan@manipal.edu

ABSTRACT

This paper predicts the average stock price for five datasets by utilizing the historical stock price data ranging from April 2009 to February 2019. Autoregressive Integrated Moving Average (ARIMA) model is used to generate the baseline, while Long Short-Term Memory (LSTM) networks is used to build the forecasting model for predicting the stock price. The efficiency of the two models is compared in terms of Mean Squared Error. The results show that the LSTM model predicts better than the ARIMA model with respect to time series forecasting. Additionally, Attention LSTM networks is employed to further study the improvement in accuracy of the stock price forecasting model.

CCS CONCEPTS

• Computing Classification System (CCS) • Computing methodologies • Machine learning • Machine learning algorithms • Ensemble methods • Boosting

KEYWORDS

Stock price, Deep Learning, Machine Learning, Time series forecasting.

ACM Reference format:

Baleshwarsingh Joosery and G Deepa. 2019. Comparative Analysis of Time-Series Forecasting Algorithms for Stock Price Prediction. In *Proceedings of 2019 International Conference on Advanced Information Science and System (AISS'19). Singapore, 6 pages.* https://doi.org/10.1145/3373477.3373699

1 Introduction

Sequence prediction problems have existed for quite some time now [1]. In the world of data science, they are regarded as one of the toughest problems to solve. Sequence prediction is inclusive of several challenges; starting with anticipating the number of shares sold moving to as far as identifying sequences from stock market sequential data, grasping the essence from a plot of a movie to speech recognition, from translating languages to anticipating the following words while texting on a smartphone.

AISS 2019, November 15–17, 2019, Singapore, Singapore

ACM ISBN 978-1-4503-7291-6/19/11…$15.00
https://doi.org/10.1145/3373477.3373699

Some people have been fascinated with the stock market for years [2]. The reason being there are too many variables that affect the value of the shares, and most people, even stockbrokers from Wall Street would admit that most of the time they do not know whether the price of a stock would drop or rise [3]. From the breakthrough in sequence prediction, came the idea to make sequence prediction profitable.

Those recent breakthroughs, especially those in the data science field, have shown that for most prediction problems, Long Short-Term Memory (LSTM) networks have a lot of potential and show promises as an efficient solution to time series forecasting problems through the various sequences they have been applied to [1].

LSTM get the edge over typical feedforward neural networks and Recursive Neural Networks (RNN) through possessing the ability of memorizing certain patterns for extended periods of time [4]. Whilst LSTMs have shown effectiveness on various problems, it is not the simplest of techniques to apply and stock prediction has always been volatile.

This paper presents a forecasting model using LSTM to predict the price of a stock. The rest of the paper is organized as follows. Sections 2 and 3 present the background and related work respectively. Section 4 discusses the proposed methodology. Section 5 explains about the implementation details and Section 6 highlights the results obtained and provides an analysis. Conclusions and future scope are discussed in the last section.

2 Background

A stock price or share price is the value that a single share has out of the many tradable stocks that a company holds. In a nutshell, the price of a single share/stock represents the maximum amount an individual is willing to buy it for, or the minimum amount that it can be purchased at.

Stock price is affected by various factors such as inflation, deflation and exchange rates. For predictive analysis, taking into account the aforementioned factors require news analytics [5] and separate algorithms which are not time series forecasting algorithms. Implementation of those algorithms and their methodologies are different from the ones being used in this experiment. As a result, to predict stock prices using only machine learning algorithms related to time series forecasting, historical data is used [6].

The features of a dataset for stock price forecasting should at least include:

- Opening price
- Highest price
- Lowest price
- Closing price

These prices show the evolution of a stock price during a stock market transaction day. Opening price is the price at which the stock starts on a trading day. Highest price and lowest price are the most expensive and the cheapest the stock has been traded at during the day. These two prices are determined at the end of a trading day. Closing price is the final price of the stock at the end of the trading day. All these prices change every trading day.

The following subsections describe the algorithms used in this experiment for the prediction of the stock price.

2.1 Recurrent Neural Networks

A major drawback for traditional neural networks is, they do not function the same way as recurrent neural networks. For example, for a classification problem such as movie plot classification, the neural network will not factor in the occurring events that led to a certain plot before classifying it. Hence, a traditional neural network is not appropriate for such problems as it is unclear how the algorithm could be used to do so.

Recurrent neural networks on the other hand do not face this issue. These networks have loops, allowing information to persist. Circulation of information from one unit of the network to the next is enabled by a loop.

One way to look at a recurrent neural network is as a duplication of the same network. Each of them sending a message to the following one.

For RNNs (recurrent neural networks), possessing a chain-like structure, shows that they are closely associated with lists and sequences. Thus, RNNs are an appropriate architecture to apply to sequential data [1].

2.2 Long Short-Term Memory

Long Short-Term Memory (LSTM) networks can be seen as a derivative of RNN for its long-term dependencies learning capabilities. To be more accurate, LSTMs are classified as sequential neural networks models, which are in turn an extension of recursive neural networks.

LSTMs have been used several times with time series data and performed satisfactorily on various types of time series forecasting problems. As a result, they are gaining popularity.

LSTMs were built to specifically tackle the problem of long-term dependency that neural networks face. Remembering and reusing information for extended time periods is their most attractive property and they have no difficulties in learning them [4].

2.3 Attention Mechanisms

Attention mechanisms in neural networks serve to orient the way inputs are processed by the model (perception) as well as memory access. A neural network takes the entire input before generating an output. However, the outputs need not be similarly affected by all the inputs. Attention filters the perceptions that can be stored in memory and filters them again on a second pass when they are to be retrieved from memory.

Attention can be given by the following: $\boldsymbol{\alpha}<t, t'>$, where $\boldsymbol{\alpha}$ is the input, t is the attention size and t' is the attention weight. To apply attention, we need to assign a value to the parameter t which will be the number of rows to focus on for each output. Then weights need to be assigned to each row depicting how much focus is to be laid to each row with respect to each other. As a result of this, now the neural network knows which rows are the most important for generating a specific output. For stock price forecasting higher weights will be set to existing records closer to the date we are predicting [7].

3 Related Work

There have been various successful implementations of time series forecasting algorithms in stock market prediction. While most models are better at predicting the trends of the stock price, some perform satisfactorily at predicting the actual opening or closing price. Forecasting the price for extended periods of time is the real challenge since stock price is affected by a lot of external and unexpected factors as well, and factors that are difficult to take into consideration in advance. The ARIMA model has a decent performance in forecasting stock price or any other form of forecasting related to time series forecasting.

Some of the well-known algorithms that have been used for predicting the stock price are Support Vector Machine (SVM) [8], Support Vector Regression (SVR) [6] and the Autoregressive Integrated Moving Average (ARIMA) model [9].

Some of the other machine learning/deep learning algorithms such as Moving Average, Linear Regression, k-Nearest Neighbors and Prophet have been used for stock price prediction. Moving Average uses the latest set of values for each prediction. In other words, for each subsequent step, the predicted values are taken into consideration while removing the oldest observed value from the set [10]. Linear Regression returns an equation that determines the relationship between the independent variables and the dependent variable [11]. k-Nearest Neighbors finds the similarity between new data points and old data points [12], and Prophet tries to capture the seasonality in the past data and works well when the dataset is large [13].

Sequence prediction is becoming more and more accurate with growing machine learning and deep learning techniques. The evolution of the price of a stock can be viewed as a sequence that is affected by various factors. The proposed model will offer the feature of offering forecasts for 20 days instead of just the next day.

4 Proposed Methodology

Stock price prediction is very volatile and is affected by various factors. However, good predictions could be obtained by using years of historical data [6]. Long Short-Term Memory (LSTM) networks is an algorithm that works well with historical data.

LSTMs are an extension of Recurrent Neural Networks (RNN), with the additional feature of memory. Thus, they are capable of remembering their inputs and learning from significant occurrences that have a considerable time lapse in between. The memory of an LSTM is similar to that of a computer and supports read, write and delete operations to the stored information [4].

Figure 1 represents the proposed methodology. The proposed methodology for predicting the stock price is divided into four minor tasks and is explained in the following subsections.

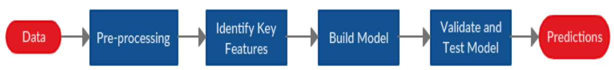

Figure 1: Methodology

4.1 Pre-Processing

In this task, the data is analyzed and then tested for outliers and missing values. Data clean up and normalization is done. At the end of the process we expect to obtain well organized data and represented in a table format.

4.2 Identify Key Features

The key features (columns) of the data that contribute to forecasting the price of the stock need to be identified. Only the required features are kept, and the rest can be regarded as superfluous and can be discarded. The dataset that we possess, after pre-processing, needs to be separated into two sets. One for training the model (i.e., training data), and the other for validating the model (i.e., validation data).

4.3 Build Model

Once the data is in good shape, we can start working on the actual model. This stage involves a lot of experimentation and discovery like testing the algorithm and more.

4.4 Validate and Test Model

At this point we ensure the final model is as good as it can be. The model performance is assessed based on the predefined quality metric (Mean Squared Error), analyze the performance of the algorithm used, tune any parameters that affect the model performance and eventually test the performance of the final model.

4.5 Assumptions and Justifications

Length of data to use as training data and how many predictions to generate: 14 months of data was used to predict the next month in a previous LSTM implementation [14]. However, this may not be the best case for stock market price forecasting. Since historical data of over 10 years old is available, the course would be to use the 10 years of data and calculate the accuracy of the predictions and reduce the amount of training data from there if required.

Feature selection: usually the "Close" feature of the dataset is used as input, and this is the feature for which predictions are made [6]. However, an individual does not buy a share of a stock at the time of closing. Therefore, to offer a new way of generating predictions, an average of all 4 price features: "Open", "High", "Low" and "Close" can be generated, that way the predicted price could represent a value at which the stock price will be at any point of the day.

5 Implementation

The implementation was done using Python libraries: Keras and Tensorflow. The datasets used for experimentation are as follows.

- AAPL (Apple.inc 2009-2019)
- GOOGL (Alphabet.inc 2009-2019)
- NKE (Nike.inc 2009-2019)
- NOK (Nokia Oyj 2009-2019)
- SNE (Sony Corp 2009-2019)

Stock price data ranging from March 2009 to March 2019 is used to predict the average stock price of the following month (datasets retrieved from Yahoo finance).

Load and visualize the dataset. Check for outliers or missing values (there should not be any, since yahoo finance is very reliable), if any found remove them or perform a mean of a set of values and replace the data.

5.1 Pre-Processing

Each dataset consists of 6 features: "Date", "Open", "High", "Low", "Close", "Adj. Close", "Volume". The "Adj. Close" and "Close" columns are exactly the same, and hence either one could be dropped. In this case, the "Adj. Close" column is dropped. Volume does not affect stock price [15]. Therefore, it can be dropped as well. The date column is set as the index of the dataset.

When making predictions "Open", "High", "Low" or "Close" can be used as the feature to predict. This paper tries to get a prediction that will be close to the actual price of the stock at any time of the day, therefore, an average of these 4 features is taken and added to the dataset under the name "OHLC".

The "OHLC" feature is used as input for the LSTM model and its values are scaled to be represented between 0 and 1. Then the model uses the training data for prediction, where each record (row) of the selected feature is taken as a single input.

5.2 Building Model

Now that the dataset is ready, the model needs to be initialized. The model is a recursive neural network and it needs to be built layer by layer.

First the model type is to be imported from the library which is a "Sequential" model. Then the following layers need to be added:

- LSTM
- Dropout (values to be discarded)
- Dense (output)

After importing the model and layers, add four LSTM layers along with the dropout regularization (to decide which values are to be dropped while making predictions). Specify the units (inputs), return sequence and input shape. Then add the Dense layer and specify units (outputs).

Compile the model and specify the optimizer (adams) and how to measure the loss (Mean Squared Error) during the training process of the model. Start the model fitting, define the inputs ("OHLC") and output ("Forecast"), epochs ("500") and batch size ("32").

5.3 Generate Output

Once the model fitting is complete, load test dataset and reshape the data as the training data used. Specify feature to be used to test predictions ("OHLC"). Get the length of the array (test data) and transform the dataset (values set between 0 and 1).

Generate predictions (output) for a specified period of time. Inverse transform the data (restore to original values from 0 to 1) and add it to the test dataset under the header "Forecast".

Plot the predictions against the actual values to visualize the performance of the model. Calculate the Mean Squared Error to get a measure of prediction accuracy.

5.4 Implementing LSTM with Attention

As discussed in Section 2.3, the steps involved in implementing LSTM with attention are as follows.

1. Define attention size *(t)*
2. Get attention weights *(t')*
3. Add attention mechanism to model (added right before model fitting)
4. Adjust parameters (set input and output streams)
5. Complete integration

6 Results and Analysis

The results obtained after applying LSTM and LSTM with Attention is presented in this section. Additionally, the ARIMA model is used on the datasets specified in Section 5, and the results obtained from the three models are finally compared.

The performance of the LSTM model, ARIMA model and Attention LSTM Model are shown in table format. A graph illustrating the forecasts against the actual values is also added. The evaluation metrics are explained and finally a conclusive decision can be made about which model performs the best.

6.1 Evaluation Metrics

Mean Squared Error (MSE):

$$\frac{1}{n}\sum_{i=1}^{n}(\text{Forecast}_i - \text{True}_i)^2 \tag{1}$$

The Mean Squared Error (MSE) takes the mean of the square of the difference between the actual values and the predicted values. The closer the value is to 0, the smaller the error.

Mean Absolute Percentage Error (MAPE):

$$\frac{100\%}{n}\sum_{i=1}^{n}\left|\frac{\text{Forecast}_i - \text{True}_i}{\text{True}_i}\right| \tag{2}$$

The Mean Absolute Percentage Error (MAPE) takes the ratio between the difference of the actual values from the predicted values to the actual values and multiplies it by 100 to give the error in percentage.

ACCURACY:

$$(100 - MAPE)\% \tag{3}$$

To find accuracy, MAPE is subtracted from 100, taking away the percentage error leaving the similarity. This gives a measure of how close the predicted values were to the actual values.

6.2 Observations

Table 1 represents the number of rows of data utilized for the prediction. By analyzing Tables 2 and 3, we can see that as lesser time period of data is considered for prediction, the results improve.

Table 1: Rows of data used

Time	10 years	5 years	1 year	6 months	3 months	1 month
Rows	2497	1259	252	124	62	20

Table 2: MSE of the models against different periods of data for the NKE stock

Model	MSE					
	10 years	*5 years*	*1 year*	*6 months*	*3 months*	*1 month*
ARIMA	1.831	1.805	1.804	1.301	1.294	3.846
LSTM	19.018	6.454	3.230	1.055	0.838	2.115
Attention	18.997	3.900	1.038	0.795	0.683	0.835

Table 3: MAPE of the models against different periods of data for the NKE stock

Model	MAPE					
	10 years	*5 years*	*1 year*	*6 months*	*3 months*	*1 month*
ARIMA	1.197	1.196	1.196	1.176	1.139	2.231
LSTM	4.141	2.435	1.826	1.031	0.907	1.498
Attention	4.653	1.807	0.921	0.879	0.839	0.904

Table 4: Average accuracy of the models for all 5 datasets against different lengths of data

Model	Accuracy in %					
	10 years	*5 years*	*1 year*	*6 months*	*3 months*	*1 month*
ARIMA	96.923	96.968	97.103	97.189	97.470	94.946
LSTM	94.642	95.924	98.793	98.893	99.170	97.876
Attention	96.223	96.933	98.963	99.045	99.365	97.895

It can also be observed from table 4 that the accuracy of the models increases by reducing the number of days considered for forecasting the stock price. It is also important to highlight that when one month of data was used to predict the price of the stock for the next month, the predictions were not as good as using three months of training data. While the exact stock price is not determined at all times, the difference between the predicted price and actual price is below 1% for most cases when three months of data (62 rows) were used for the Attention LSTM model.

The justification for this is that the LSTM model can learn very long-range connections in a sequence and past data (especially those more than a year old) does not help much in determining the price of a stock in upcoming days as there may have been some external factors that affected the price during that period.

The ARIMA model however is not greatly affected by the amount of training data used. It performs better than LSTM for data over one-year-old. This is because the ARIMA model does not remember connections in a sequence, but mostly generates predictions through an automated mean mechanism.

The testing of the models stopped at one month of data (20 rows) as the results actually worsened from three months (62 rows). Since we are predicting the prices for the next month it seems that a single month of training data is not enough for any of these models to find the correlations that they did from three months of data.

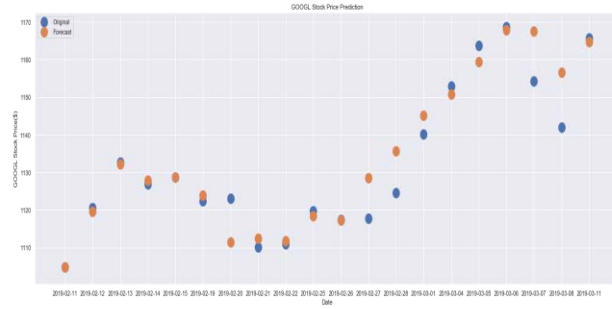

Figure 2: Forecasts of GOOGL stock using 3 months of data

Figure 2 shows the forecasted prices for the GOOGL stock using three months of training data against the actual price on a scatter plot. We can see that the prices are not far apart especially for the next day prediction. The graph also reveals that the Attention LSTM model is able to capture the trends of the prices when making the predictions.

6.3 Gained Insights

- To get better predictions, use the most recent data available. The results obtained are satisfactory when a lesser time span of three months of the past data is used as training data. The result worsens when a larger time span of data is used.
- The LSTM model can be improved by integrating some additional mechanisms to it as in this case the attention mechanism.

6.4 Final Outcome

The average accuracy obtained for each of the models is as follows: 96.766% for the ARIMA Model, 97.549% for LSTM and 98.070% for Attention LSTM.

Table 5: MSE of the models against each dataset for three months

Model	MSE				
	AAPL	GOOGL	NKE	NOK	SNE
ARIMA	51.245	1446.835	1.294	0.069	4.916
LSTM	4.393	180.194	0.838	0.003	0.263
Attention	2.101	96.654	0.683	0.001	0.134

Table 6: MAPE of the models against each dataset for three months

Model	MAPE				
	AAPL	*GOOGL*	*NKE*	*NOK*	*SNE*
ARIMA	3.537	2.884	1.139	3.717	4.106
LSTM	0.847	0.866	0.907	0.754	0.772
Attention	0.609	0.700	0.839	0.488	0.627

The models performed the best on all the datasets when three months of data are used for training. Tables 5 and 6 show the MSE and MAPE respectively for all the five datasets. The MSE values for the GOOGL stock are comparatively high to the other stocks because the price of a single share is about $1180 and most of the others are less than a $100 per share. The MAPE values, shows less than 1% error for the LSTM and Attention LSTM for all the datasets.

7 Conclusion and Future Scope

The ARIMA model performs better than the other models on large data samples (over one year of data). Smaller data samples lead to better forecasts on every model. The Attention mechanism improves the accuracy of the LSTM Model and outperforms the ARIMA Model for datasets holding one years' worth of data or less. To make optimal forecasts using three months of data is ideal.

The Attention mechanism improved the predictions generated by the LSTM model, and it is worth trying other mechanisms with the LSTM model and see how they affect the predictions.

The LSTM model is a much more powerful model than most machine learning algorithms. There are new models that have been developed after the LSTM model such as the Gated Recurrent Unit (GRU) and these models should be implemented for this same problem.

As there is more and more progress being made in the machine learning and deep learning fields, the new models being developed could be tested against the LSTM model to compare their performance. This could help in generating more accurate predictions over longer periods of time.

Furthermore, in this experiment, external factors such as market news and market sentiment were not taken into account. With new techniques such as news analytics [5] and sentiment analysis [16], taking such factors into consideration, although difficult, is possible. Being able to capture such information and incorporating them in the prediction model can further boost the model and give even better results.

REFERENCES

[1] Samy Bengio, Oriol Vinyals, Navdeep Jaitly, Noam Shazeer (2015). Scheduled Sampling for Sequence Prediction with Recurrent Neural Networks. Advances in Neural Information Processing Systems 28 (NIPS 2015), Mountain View, CA, USA.

[2] Louis M. Rotando & Edward O. Thorp (1992). The Kelly Criterion and the Stock Market. The American Mathematical Monthly, vol. 99:10, pp. 922-931.

[3] Donald C. Langevoort (1996). Selling Hope, Selling Risk: Some Lessons for Law from Behavioral Economics about Stockbrokers and Sophisticated Customers. California Law Review, vol. 84, no. 3.

[4] Sepp Hochreiter and Jürgen Schmidhuber (1997). Long Short-Term Memory. Neural Computation, vol. 9, no. 8, pp. 1735-1780.

[5] Axel Groß-Klußmann, Nikolaus Hautsch (2011). When machines read the news: Using automated text analytics to quantify high frequency news-implied market reactions. Journal of Empirical Finance, Volume 18, Issue 2, pp. 321-340.

[6] Xianxian Hou, Shaohan Zhu, Li Xia, Gang Wu (2018). Stock Price Prediction based on Grey Relational Analysis and Support Vector Regression. The 30th Chinese Control and Decision Conference, Shenyang, China.

[7] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, Illia Polosukhin (2017). Attention is All you Need. Advances in Neural Information Processing Systems 30, California.

[8] Hakob Grigoryan (2017). Stock Market Trend Prediction Using Support Vector Machines and Variable Selection Methods. International Conference on Applied Mathematics, Modelling and Statistics Application, Beijing, China.

[9] Ayodele A. Adebiyi, Aderemi O. Adewumi, Charles K. Ayo (2014). Stock Price Prediction Using the ARIMA Model. UKSim-AMSS 16th International Conference on Computer Modelling and Simulation, Cambridge, United Kingdom.

[10] T. Kimoto, K. Asakawa, M. Yoda, M. Takeoka (1990). Stock market prediction system with modular neural networks. IJCNN International Joint Conference on Neural Networks, San Diego.

[11] Erdinç Altay, Satman M Hakan (2005). Stock Market Forecasting: Artificial Neural Network and Linear Regression Comparison in an Emerging Market. Journal of Financial Management & Analysis, vol. 18, no. 2, pp. 18-33.

[12] Khalid Alkhatib, Hassan Najadat, Ismail Hmeidi, Mohammed K. Ali Shatnawi (2013). Stock Price Prediction Using K-Nearest Neighbor (kNN) Algorithm. International Journal of Business, Humanities and Technology, vol. 3, no. 3.

[13] Shruti Chouksey (2019). Stock price prediction using time series models. Dublin Business School, Dublin, 2019.

[14] Chen Xiaochen, Wei Lai, Xu Jiaxin (2017). House Price Prediction Using LSTM. The Hong Kong University of Science and Technology, Hong Kong.

[15] John Y. Campbell, Sanford J. Grossman, Jiang Wang (1993). Trading Volume and Serial Correlation in Stock Returns. The Quarterly Journal of Economics, vol. 108, no. 4, pp. 905-939.

[16] Maite Taboada, Julian Brooke, Milan Tofiloski, Kimberly Voll, and Manfred Stede (2011). Lexicon-Based Methods for Sentiment Analysis. Computational Linguistics, 37:2, pp. 267-307.