Department of Artificial Intelligence and Machine Learning
School of Computer Science & Engineering

A Report

On

STOCK PRICE PREDICTION USING LSTM WITH VARYING SEQUENCE LENGTHS

carried out as part of the course: AI3132.
Submitted by

Devanshi Kumar

Registration No.: 219310324

AIML-V

in partial fulfilment for the award of the degree

of

BACHELOR OF TECHNOLOGY

In

Computer Science and Engineering (AIML)

Manipal University Jaipur-303007.
November, 2023.

Acknowledgement

This project would not have completed without the help, support, comments, advice, cooperation and coordination of various people. However, it is impossible to thank everyone individually; I am hereby making a humble effort to thank some of them.

I acknowledge and express my deepest sense of gratitude of my internal supervisor Dr. Deepak Panwar for his constant support, guidance, and continuous engagement. I highly appreciate his technical comments, suggestions, and criticism during the progress of this project.

I owe my profound gratitude to Mr. Sandeep Chaurasia, Head, Department of CSE AI-ML, for his valuable guidance and facilitating me during my work. I am also very grateful to all the faculty members and staff for their precious support and cooperation during the development of this project.

Finally, I extend my heartfelt appreciation to my classmates for their help and encouragement.

Thank You.

Department of Computer Science and Engineering
School of Computing & Information Technology

Date: November 21, 2023.

CERTIFICATE

This is to certify that the project entitled “*Stock Price Prediction Using Lstm With Varying Sequence Lengths*" is a bonafide work carried out as ***Project Based Learning (Course Code: AI2270)*** in partial fulfillment for the award of the degree of Bachelor of Technology in CSE-AIML, under my guidance by ***Devanshi Kumar*** bearing registration number 219310324, during the academic semester *V of year 2022-23.*

Place: Manipal University Jaipur, Jaipur.

Name of the project guide: Dr. Deepak Panwar

Signature of the project guide: ______________________

Contents

1. Abstract

Predicting stock prices has been a complicated and unpredictable task. Accurate forecasts are vital for successful investments, as they can help investors make informed decisions to minimize risks and optimize returns. Neural network methods have gained popularity in recent years because of their ability to handle large datasets and extract intricate patterns. This study uses a long short-term memory (LSTM), a neural network architecture, to predict the closing price of S&P 500 index. A neural network is constructed with LSTM layers for time series prediction and the model is experimented with various sequence lengths. The performance is compared using standard assessment metrics –Root Mean Square Error (RMSE), Mean Absolute Error (MAE), and R-squared (R2). The experimental results show that shorter sequence lengths provide better results in capturing short-term fluctuations.

Keywords: LSTM; Stock price prediction; Neural Networks; Sequence length

2. Introduction

The fluctuations in the stock price are unpredictable because of several factors such as changes in the unemployment rate, changing monetary policies of countries, natural disasters, public health conditions, and several others. The goal is to maximize profits while minimizing risks.

Stock markets are naturally noisy, non-parametric, non-linear, and deterministic chaotic systems (Ahangar, Yahyazadehfar, & Pournaghshband, 2010). Feature selection also poses a problem in the prediction. There has been a trend in which some researchers use only technical indicators, whereas others use historical data (Di Persio and Honchar, 2016, Kara et al., 2011, Nelson et al., 2017, Patel et al., 2015, Qiu and Song, 2016, Wang and Kim, 2018).

Over the years, many predictive models have been developed to address the volatility and complexity of financial markets. Machine learning algorithms such as Moving Averages, Support Vector Machines (SVM), and Random Forest have served as prominent tools for stock price prediction. However, these models may encounter limitations when dealing with the intricate dynamics of financial markets.

This study makes use of Long Short-Term Memory (LSTM) model, a specialized type of Recurrent Neural Network (RNN) to predict the stock index price. In an RNN architecture, information is being passed from one timestep to the next internally within the network, without retaining past information. In contrast to basic feedforward neural networks, LSTM overcomes the vanishing gradient problem (Hochreiter, 1998) by memorizing the information for a longer period.

As outlined in the diagram, data is first collected and Exploratory Data Analysis (EDA) is performed on the dataset. The data is then normalized using the min-max normalization technique. The Input sequence for the LSTM model is created using a specific time step. The hyperparameters such as number of epochs, learning rate, batch size, and time step have also been incorporated in the model. The model is trained with early stopping callback and predictions are made. The quality of the proposed model is assessed through RMSE, MAE, and R2 value, and compared with the predictions made when the sequence length varies.

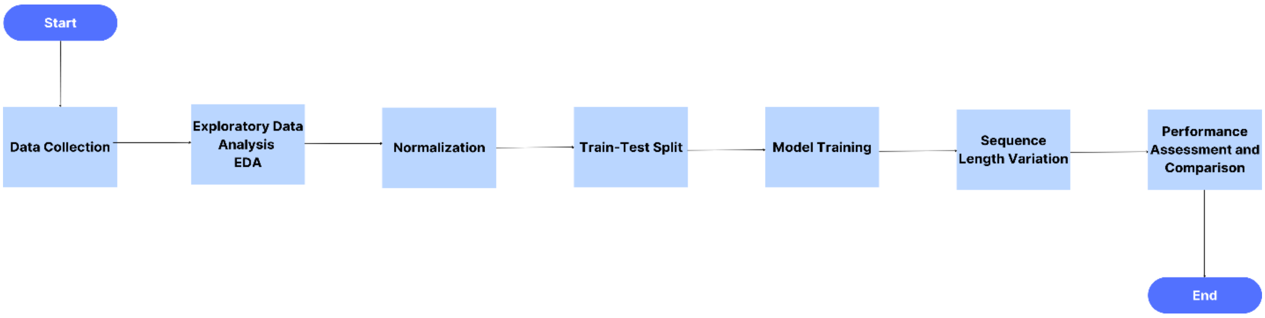

Start
Data Collection
Exploratory Data Analysis EDA
Normalization
Train-Test Split
Model Training
Sequence Length Variation
Performance Assessment and Comparison
End

Fig. 1. Schematic diagram of the proposed research framework

3. Long Short-Term Memory (LSTM)

Long short-term memory (LSTM) is a variation of an RNN model. An RNN can only memorize short-term information, but LSTM can handle long time-series data. An LSTM model has automatic control for maintaining pertinent features in the cell state and is capable of recalling prior long-term time-series data.

The dimension of the data determines how many nodes are in the input layer of a neural network with a single hidden layer. The input layer nodes are connected to the hidden layer through connections referred to as 'synapses.' A weight coefficient, which is linked to every connection between two nodes from the input to the hidden layer, is essential to signal processing. These weights are automatically adjusted during the learning process so that, by the end of the learning phase, the artificial neural network (ANN) has the ideal weights for every synapse.

The nodes in the hidden layer apply an activation function, typically a sigmoid or hyperbolic tangent (tanh) function, to the weighted sum from the input layer. This activation function commonly employs the SoftMax function to turn the input into values intended to minimizing the error between the training and testing data.

The neural network's output layer is made up of the values that come from this transition. These values, however, may not represent the best possible outcome. In such instances, a backpropagation procedure is used to find the best error value. This backpropagation mechanism connects the output layer with the hidden layer, sending signals to modify the weights in order to achieve the ideal configuration for minimizing error. This iterative procedure is done to improve forecasts and reduce prediction errors.

Just like other neural networks, recurrent Neural Networks (RNN) have weights, biases, layers, and activation functions. The major difference is that Recurrent Neural Networks also have feedback loops which make it possible to use sequential input values, like stock market prices, to make predictions. Unlike feedforward neural networks, RNNs have connections that loop back on themselves, allowing them to maintain a hidden state that remembers information from previous time steps. Gradients are computed and propagated from the output layer back to the input layer in an RNN during the backpropagation process to update the network's weights. However, when the network is deep or when it encounters activation functions with gradients less than 1, these gradients can become extremely small as they are multiplied together layer by layer. This is the vanishing gradient problem. As a result, these early layers fail to learn long-term dependencies and are unable to capture relevant information from distant time steps in sequential data. Conversely, when gradients are greater than 1 and are multiplied during backpropagation, they can grow exponentially as they are passed backward through the layers This can cause the optimization process to become unstable, as weight values can become extremely large or small.

To address this issue, more advanced RNN architectures like Long Short-Term Memory (LSTM) have been developed, which overcome the limitations of traditional RNNs when dealing with sequential data. They achieve this through a complex structure of interconnected memory cells, each equipped with gating mechanisms.

The set of cells responsible for storing and managing the flow of prior data streams is a critical component in the architecture of an LSTM node. Each cell is connected to the one before it by an upper line, allowing historical data to be transferred into the present cell. The independence of these cells is critical since it allows for selective filtering or value addition between them. A layer of sigmoidal neural networks known as gates is used to effectively govern the state of each cell.

The gates are crucial in managing the flow of information within the cells:

The Forget Gate produces an output ranging from 0 to 1, where 1 signifies "retain this information entirely," while 0 indicates "completely disregard this."

The Memory Gate, consisting of an "input door layer" followed by a tanh layer, determines which new data should be incorporated into the cell. The "input door layer" employs a sigmoid function to select the values to be updated, and the tanh layer generates a vector of potential new values that can be added to the cell's state.

The Output Gate decides what information will be the ultimate output of each cell. It effectively determines what information is passed to the next node in the sequence.

4. Methodology

4.1 Exploratory Data Analysis

4.1.1 Data Collection: S&P 500 stock price is downloaded using yfinance. The date ranges from 2019 to 2023.

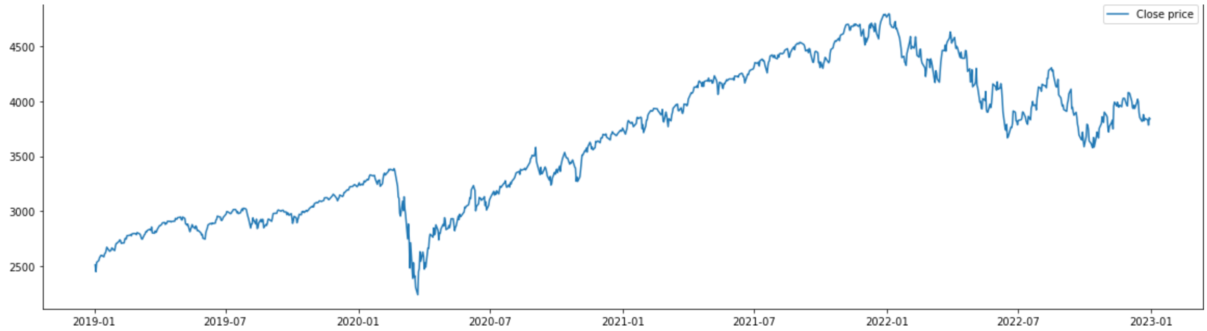
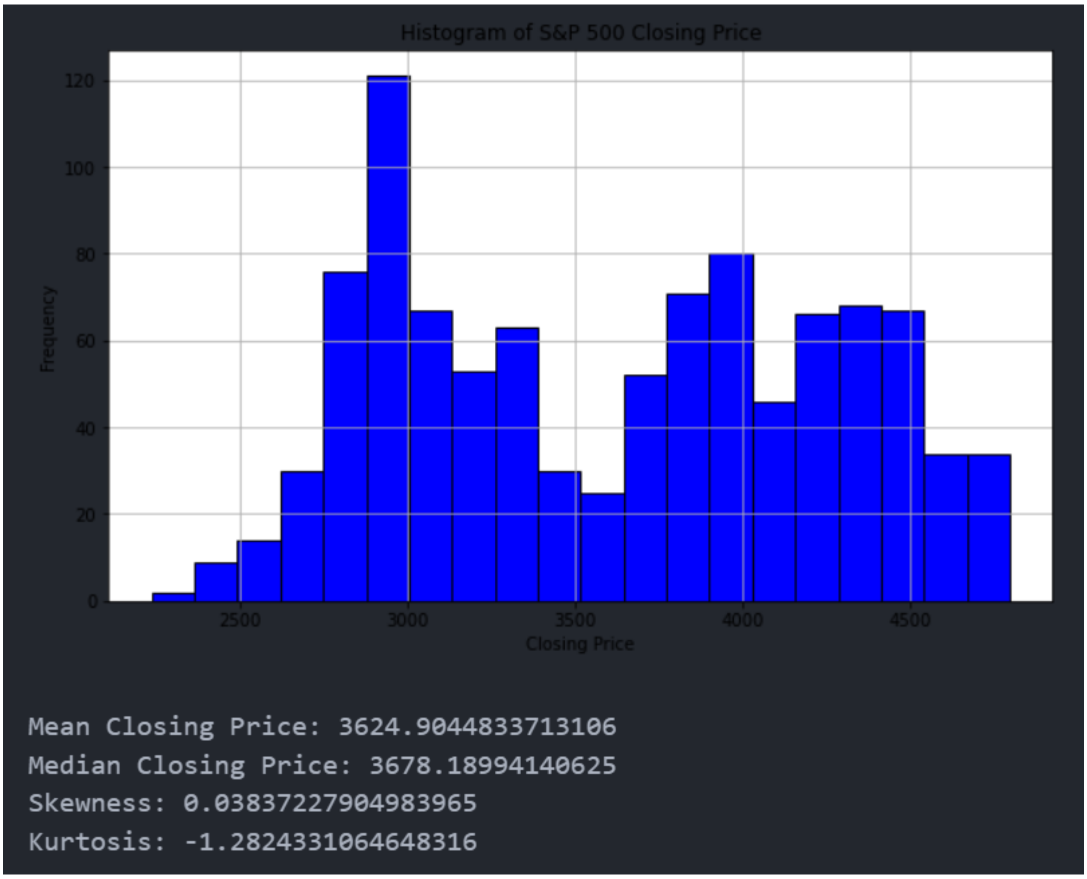


Fig 2: Closing Price for the stock index

4.1.2 Insights:



```
Mean Closing Price: 3624.9044833713106
Median Closing Price: 3678.18994140625
Skewness: 0.03837227904983965
Kurtosis: -1.2824331064648316
```

Fig 3: Histogram of closing prices

The mean closing price is lower than the median, indicating that there may be some relatively lower closing prices that are pulling the mean down.

The negative kurtosis suggests that the distribution has lighter tails and fewer extreme values compared to a normal distribution.

4.1.3 Data Cleaning and Preprocessing: This involves data cleaning, handling missing values, and ensuring the dataset is in a suitable format for analysis. Detection for outliers is also applied to enhance the quality of the dataset.

4.1.4 Time Series Analysis (Trend, Seasonality, Residual):

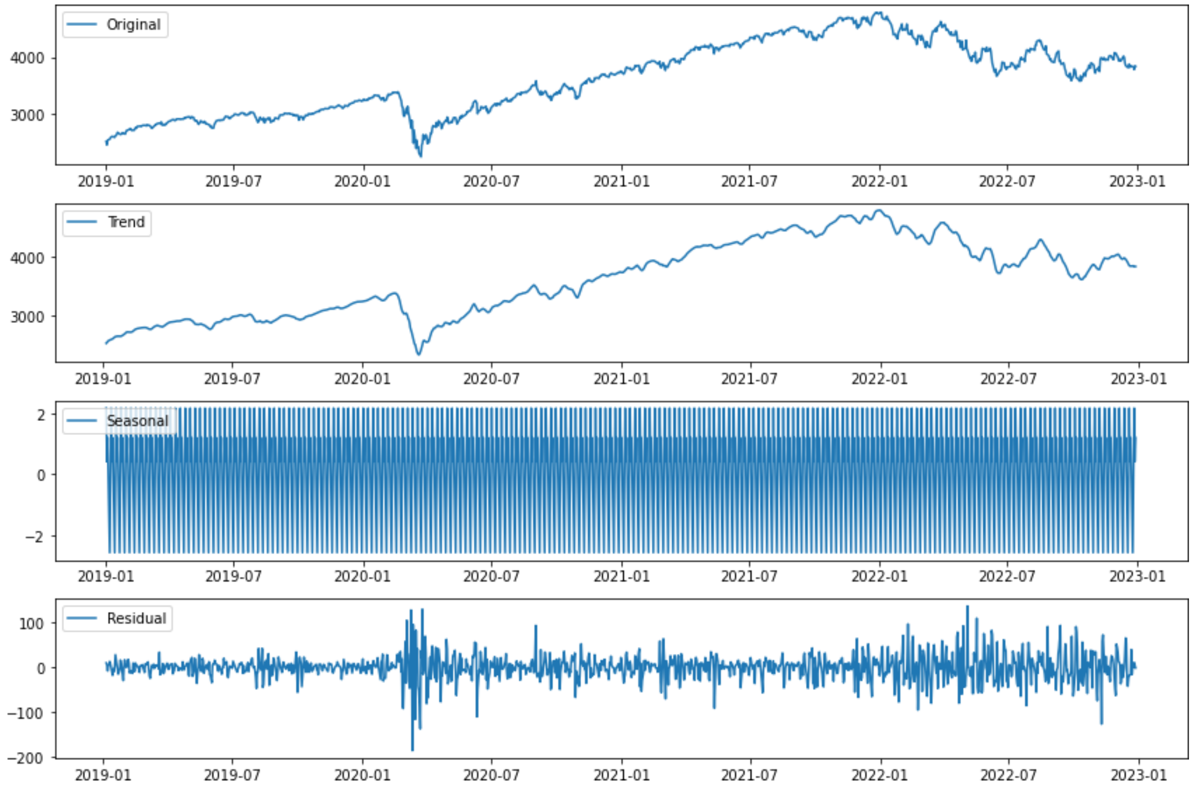


Fig 4: Graph for trend, seasonality and residual

The trend component represents the long-term movement or general direction in the data. It captures the underlying, gradual changes in the time series over a more extended period. The seasonality component reflects the repeating patterns or fluctuations in the data that occur at fixed intervals, typically within a year or a shorter time frame. The residuals, also known as errors or noise, are the leftover variations in the time series data that cannot be explained by the trend or seasonality components.

4.2 LSTM Model

4.2.1 Data Preparation: The dataset is subjected to normalization using Min-Max scaling to ensure that all features fall within the same range, thereby aiding the training process. Subsequently, the data is partitioned into training and testing sets, with the training set encompassing 80% of the dataset. Notably, for the testing data (X_test), a specific sequence of the last 200 data points is retained for predictive analysis.

4.2.2 Model Architecture: A model with two LSTM layers is being used. The structure of the model is formulated using the Keras Sequential API. Each LSTM layer comprises 50 units and employs the Rectified Linear Unit (ReLU) activation function. Notably, the 'return_sequences=True' attribute is set for all LSTM layers, except for the final layer, to facilitate the propagation of sequences. The model concludes with a Dense layer consisting of a single unit. The mean squared error is chosen as the loss function, and the 'adam' optimizer is utilized to minimize this error.

4.2.3 Early Stopping: To mitigate the risk of overfitting, the model incorporates an early stopping mechanism. This involves monitoring the validation loss and, in the event of a lack of improvement, restoring the model's weights to their optimal configuration. The early stopping callback enhances the generalization capabilities of the model by preventing excessive training.

4.2.4 Model Training: The LSTM model is subsequently trained using the training dataset. The number of epochs and the batch size are selected to meet the specific requirements of the study. during training, the model sees the entire training dataset in 50 complete iterations (epochs), and for each iteration, it processes 32 samples at a time (mini-batches).

4.2.5 Making predictions: After successful training, the LSTM model is employed to make predictions on the test dataset, thereby assessing its proficiency in accurately forecasting stock prices.

4.3 Varying Sequence Lengths

4.3.1 Sequence Length Variation: A series of LSTM models is trained, each with a distinct sequence length. Sequence lengths of varying granularity are considered, encompassing both shorter and longer intervals of historical data.

4.3.2 Model Reconfiguration: For each LSTM model, the sequence length is adjusted in accordance with the specific configuration. Hyperparameters such as the number of LSTM layers, batch size, and other settings remain consistent to maintain experimental integrity.

4.3.4 Performance Assessment: The performance of each LSTM model, with its respective sequence length, is assessed using suitable evaluation metrics. Key metrics such as Root Mean Squared Error (RMSE), Mean Average Error (MAE), and R-squared (R2) are employed to gauge the predictive accuracy.

4.3.5 Optimal Sequence Length: The sequence length that results in the most precise and reliable predictions is identified as the optimal choice. This determination is based on a comparative analysis of the performance metrics across varying sequence lengths.

The comparison of different sequence lengths provides useful insights into the LSTM model's adaptability to diverse temporal dependencies and aids in identifying the most efficient sequence length for stock price prediction.

This enhanced methodology enables a thorough analysis of the LSTM model's performance under various scenarios and provides assistance for picking the sequence length that best fits with the dataset and prediction aims.

5. Experiment and Results

After the data is normalized, the dataset is reshaped and split into the training and testing data sets. The goal is to predict the closing price of S&P 500 index with high accuracy.

5.1 Model performance metrics

LSTM architecture is implemented to predict the closing price with varying sequence lengths. Prediction accuracy and reliability of the model is assessed by calculating three different performance metrics – RMSE, MAE, and R2.

RMSE measures the average magnitude of prediction errors in terms of the stock price while MAE measures the average absolute difference between predicted and actual stock prices. R2 assesses the goodness of the fit by quantifying the proportion of variance in stock prices that the model explains. A model with the smallest RMSE and MAE along with the greatest possible R2 would be considered as the best model.

The experiment uses python programming environment along with Tensorflow and Keras APIs. The machine configurations are stated in the table:

Machine configuration	**VSCode Jupyter notebook**
Environment	Python 3.8.5, TensorFlow, and Keras APIs
Architecture	LSTM model

Table 1: Computing environmental condition

5.2 Result Summary

The results of the experiments are summarized below:

Sequence Length	**RMSE**	**MAE**	**R2**
10	0.033997	0.023155	0.980872
20	0.031421	0.022568	0.983059
50	0.024309	0.016855	0.990101
100	0.032984	0.024781	0.981771
150	0.029291	0.022009	0.981535

Table 2: Model performance metrics using varying sequence lengths

The results reveal a clear pattern in the model’s performance as the sequence length varies. From the above table, sequence length of 50 is optimal as it provides us with the least MSE and MAE values while keeping the R2 score nearer to 1.

The choice of a sequence length of 50 strikes a balance between capturing short-term and long-term trends in stock prices. It allows the model to consider a sufficiently extended historical context while still being responsive to short-term market changes. This is recommended for achieving accurate and reliable predictions, particularly when both minimizing prediction errors (MSE and MAE) and maintaining a strong model fit (R2).

6. Conclusion

The effect of various sequence lengths in an LSTM-based model for stock price prediction is calculated in this study. The experiments yielded important insights into the relationship between sequence length and model performance. This investigation shows that the length of the sequence is a pivotal factor in optimizing the accuracy and reliability of stock price predictions.

7. References

Adil Moghar, Mhamed Hamiche, Stock Market Prediction Using LSTM Recurrent Neural Network, Procedia Computer Science, Volume 170, 2020

Hum Nath Bhandari, Binod Rimal, Nawa Raj Pokhrel, Ramchandra Rimal, Keshab R. Dahal, Rajendra K.C. Khatri, Predicting stock market index using LSTM, Machine Learning with Applications, Volume 9, 2022

Pengfei Yu, Xuesong Yan, Stock Price prediction based on deep neural networks, 2018

Ahangar, Yahyazadehfar, & Pournaghshband, The Comparison of Methods Artificial Neural Network with Linear Regression Using Specific Variables for Prediction Stock Price in Tehran Stock Exchange , 2010