International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

Stock Price Prediction

Mr. Shreenidhi B S[1], Shirisha S[2], Thejaswini K P[3], Vismitha K[4], Sowmya N[5]

[1] Dayananda Sagar Academy of Technology And Management shreenidhibs@dsatm.edu.in
[2] Dayananda Sagar Academy of Technology And Management shirishareddy128@gmail.com
[3] Dayananda Sagar Academy of Technology And Management thejaswinikp2k4@gmail.com
[4] Dayananda Sagar Academy of Technology And Management vismithashetty24@gmail.com
[5] Dayananda Sagar Academy of Technology And Management sowmyareddy9380@gmail.com

ABSTRACT –

It is an important function for the investors and financial analysts to predict stock prices to know about the future trend of markets, although the nature of volatility of the financial markets does not help it easily. In this study, the possibility of applying the use of LSTM networks in Recurrent Neural Network (RNN) has been taken into account, predicting the stock price of Google Inc. with 20 years of data from Yahoo Finance. The technical indicators used include 100-day, 200-day, and 250-day Moving Averages to enhance the prediction accuracy. Data preprocessing was done by normalizing the dataset using Min-Max scaling and splitting the data into 70% for training and 30% for testing. Performance was measured using RMSE, and it was found that the model was able to capture overall trends but failed to capture short-term fluctuations due to market volatility. This research focuses on the ability of LSTM networks to predict future stock prices and indicates open issues for further improvement of the predictive performance.

Key Words - Stock price prediction, LSTM, Streamlit, financial forecasting, time series, machine learning, moving average, Yahoo Finance API, interactive visualization.

Introduction :

The stock market has been a highly intriguing yet intricate subject matter for researchers and practitioners over the years. Predicting stock prices, or, to be completely borne out, stocks, is a daunting task-an exercise in folly-easy as it appears. Factors like economic indicators, political happenings, investor sentiment, and market psychology act upon the stock prices to convolute their prediction. Some statistical models like ARIMA or linear regression have often been seen as incapable of modeling the complex non-linear relationships in financial data.

In recent years, deep-learning models, above all, recurrent neural networks, have emerged as powerful tools for sequential data analysis. Whereas, long short-term memory(the LSTM) networks serve as a variant of recurrent neural networks designed to deal with the long-term dependency problem in sequential data. The LTSM is intended to explore the suitability of prediction in the study when based on historical stock price pattern everywhere.

This paper specifically takes up stock prices of Google (GOOG) from Yahoo Finance to apply LSTM networks for price prediction. Here, the research aims to show that LSTM can do a good job of explanatory modeling underlying stock trends while keeping into focus how these variations can be thrown off by the stock market fluctuations. The paper considers tasks connected to LSTM model design, training, and performance evaluation and discusses the findings in practical real-world trading system applications

Related Work :

Stock price prediction has been widely studied, with various machine learning models applied to forecast prices. Traditional methods, such as ARIMA and linear regression, have been popular for time series forecasting, but their limitations in capturing non-linear relationships have led to the adoption of more advanced techniques. Among these, machine learning methods like Support Vector Machines (SVM) and Random Forests have been used, but they still fall short when dealing with large datasets with temporal dependencies.

Deep learning methods, particularly LSTMs, have recently shown great promise in the financial forecasting domain. LSTM networks are designed to handle sequential data by retaining long-term dependencies, making them well-suited for tasks such as stock price prediction, where the price at any given time depends on previous prices and market trends. Several studies have demonstrated that LSTM networks outperform traditional machine learning models, especially when trained on large amounts of historical data.

Some works have also explored hybrid models, combining LSTM with other algorithms like Genetic Algorithms (GA), Particle Swarm Optimization (PSO), or ARIMA to enhance the model's ability to make more accurate predictions. These hybrid approaches aim to optimize the model's hyperparameters or integrate external features for better performance.

This paper builds on previous research by applying a standalone LSTM model to predict stock prices, using a combination of moving averages and price data from Google stock, with an evaluation based on RMSE.

Code:

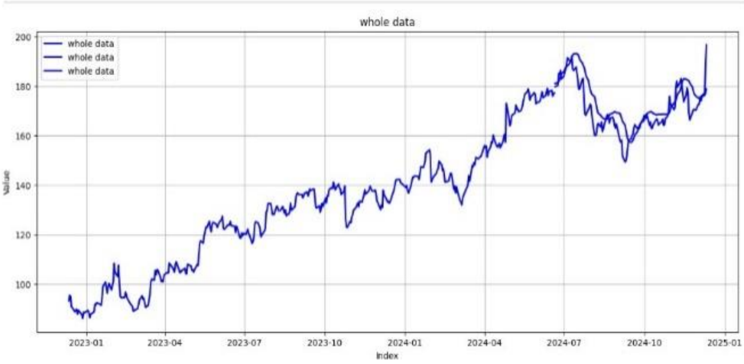
```
google_data['MA_for_250_days'] = google_data['Adj Close'].rolling(250).mean()
google_data['MA_for_200_days'] = google_data['Adj Close'].rolling(200).mean()
google_data['MA_for_100_days'] = google_data['Adj Close'].rolling(100).mean()
```

Data Collection and Processing :

3.1 Data Collecting

The data required for this study was extracted through the use of the Yahoo Finance Python library, yfinance; widely used for retrieving history of market data. The number of rows in the database exceeds 20 years; all Google Inc. stock prices are included from the year 2004 up until now. There are six significant columns: Open, High, Low, Close, Adjusted Close, and Volume. Among these, it was deemed that the price one would give more weight on would be the Adjusted Close which considers dividends and splits given to determine the stock. Hence, with this approach in mind, it becomes entirely adjusted over time as suitable for analyzing long-term trend analysis, predictive modeling.

Price	Adj Close	Close	High	Low	Open	Volume
Ticker	GOOG	GOOG	GOOG	GOOG	GOOG	GOOG
Date						
2022-12-12 00:00:00+00:00	93.225029	93.559998	93.875000	91.900002	93.089996	27380900
2022-12-13 00:00:00+00:00	95.506828	95.849998	99.800003	95.379997	98.070000	34788500
2022-12-14 00:00:00+00:00	94.968758	95.309998	97.220001	93.940002	95.540001	26452900
2022-12-15 00:00:00+00:00	90.873482	91.199997	94.029999	90.430000	93.540001	28298800
2022-12-16 00:00:00+00:00	90.534698	90.860001	91.750000	90.010002	91.199997	48485500



Data Preprocessing

The data was subjected to several critical preprocessing steps to prepare it for predictive modeling. First, the dataset was checked for missing values using the `.isna().sum()` function, and no gaps were found, so imputation was not required. Then, feature engineering was applied by calculating 250-day, 200-day, and 100-day moving averages, capturing long-term, medium-term, and short-term price trends to provide insights into market momentum and potential trading signals. The efficiency for training the model could further be improved by scaling **Adjusted Close** prices using the `MinMaxScaler` function, which is available within `sklearn.preprocessing`. Values should be scaled between 0 to 1. After finishing all the preprocessing, data was split into training at 70% and the test at 30%, to have real performance of the model through its capabilities on unseen data that give unbiased results.

This involved some key steps in getting the data ready for use with model training:

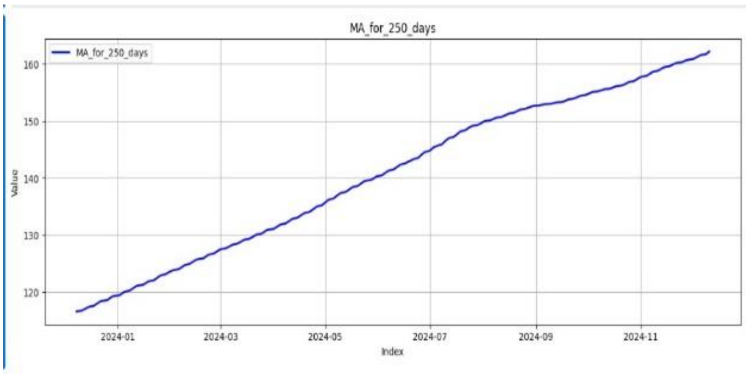
Missing Data Treatment:

Verified for missing values using .isna().sum(). There was no missing data found. Thus, imputation is not needed.

Price	Ticker	
Adj Close	GOOG	0
Close	GOOG	0
High	GOOG	0
Low	GOOG	0
Open	GOOG	0
Volume	GOOG	0

Feature Engineered:

Added 250-day, 200-day, and 100-day moving averages to capture long-term, medium-term, and short-term stock price trends.



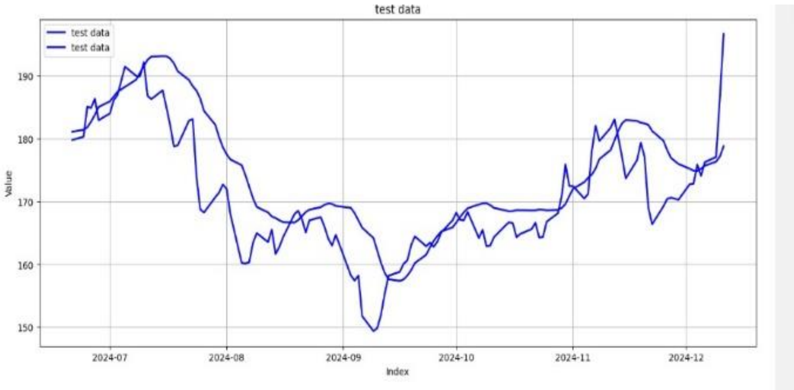
Data Scaling:

Normalized the Adjusted Close prices using MinMaxScaler to scale values between 0 and 1 for efficient model training.

Train-Test Set Split:

Split the data into 70% for training and 30% for testing to evaluate the model's performance on unseen data.

Date	Original_test_data	Predictions
2024/06/21 00:00:00+00:	179.819016	181.149460
2024/06/240 0:00:00+00:	180.347717	181.424591
2024/06/25 00:00:00+00:	185.126007	181.838562
2024/06/26 00:00:00+00:	184.916504	182.744415
2024/06/27 00:00:00+00:	186.402878	183.826874



Methodology :

4.1 Model Architecture:

In this work, the authors employed a long short-term memory (LSTM) network, which is a type of recurrent neural network (RNN) that is capable of learning from historical data and retaining long memories. The formulation of the LSTM model was as follows: LSTM model formulation:

Input Layer:

A model input consists of 3D arrays of stock prices, where the dimensions are number of samples, number of time steps and number of features. Each of these sequences is 100 sequences of daily stock prices. For every sequence, the LSTM model forecasts the stock price the next day.

LSTM Layers:

- ➢ The first LSTM layer, the one with 128 units and which returns sequences, is connected to the second layer, which is the second LSTM layer and it contains 64 units and does not return sequences.
- ➢ The second LSTM layer contains 64 units and does not return sequences. It feeds into a Dense layer.

Dense Layers:

The LSTM outputs are handled by the 25 units Dense layer, and the expected stock price is handled by the last output layer which has a single unit.

Compilation:

The model was implemented in conjunction with the Adam optimizer and the Mean Squared Error (MSE) loss function. The Adam optimizer is a common optimizer for many people as it is effective for deep learning model training while MSE is a common loss function for regressions.

Training:

In terms of epochs per batch, there were two cycles each consisting of a batch size of 1. As a result, the timeseries aspect of the data enables the model to recognize patterns and sequentially adjust weights to achieve minimum prediction error during the training phase.

Model Training and Prediction:

The training procedure consisted of LSTM model training over the training data. After training, the model was able to forecast the stock's price for the testing set. In order to facilitate a comparison of the predicted stock prices and the actual prices, the stock price estimation obtained from the model was inverse transformed to the original scale using the scaler.

- **Code Example for Training:**

model.fit(x_train, y_train, batch_size=1, epochs=2)

- **Code Example for Prediction:**

predictions = model.predict(x_test)
inv_predictions = scaler.inverse_transform(predictions)
inv_y_test = scaler.inverse_transform(y_test)

Evaluation Metrics :

The performance of the model was evaluated using the Root Mean Squared Error (RMSE) metric, which measures the average magnitude of0 the errors between predicted and actual values. The formula for RMSE is:

RMSE=n1i=1∑n(yi−y^i)2

Where:

- yi is the actual stock price.
- y^i is the predicted stock price.

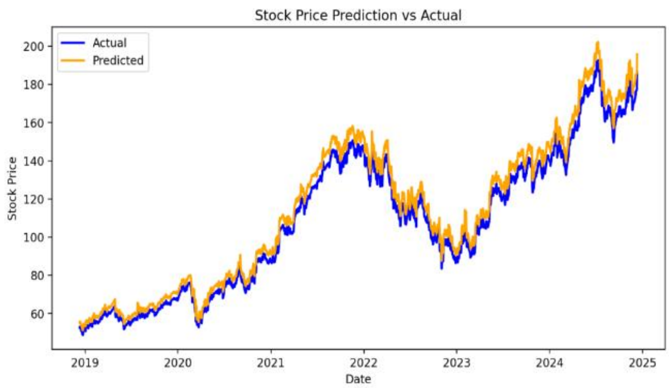
RMSE is a common evaluation metric for regression tasks, where smaller values indicate better performance.

Results :

The resulting model performance was assessed by RMSE with regard to the test dataset. The particular results came back as follows:

- RMSE: 20.47 (It should be noted that this figure will change as the share varies and the model is evaluated)
- The RMSE value suggests that while the model captures the long term direction of the stock price, it struggles to forecast short term instantaneous changes that typify financial markets. This was to be expected as the price of stocks is determined by a host of variables a good number of which do not get captured by historical price data alone.

	Adj Clos	Close	High	Low	Open	Volume
Date	GOOG	GOOG	GOOG	GOOG	GOOG	GOOG
2004-12-13 00:00:00	4.2301	4.2453	4.3133	4.2204	4.2882	193,466,452
2004-12-14 00:00:00	4.4346	4.4506	4.4538	4.2242	4.259	445,198,482
2004-12-15 00:00:00	4.4617	4.4777	4.5004	4.4	4.4331	460,559,845
2004-12-16 00:00:00	4.3795	4.3953	4.4954	4.3823	4.4072	344,197,318
2004-12-17 00:00:00	4.4691	4.4852	4.4956	4.3973	4.4025	296,555,412
2004-12-20 00:00:00	4.5917	4.6082	4.6939	4.5298	4.533	394,854,485
2004-12-21 00:00:00	4.5602	4.5766	4.6795	4.5679	4.6404	221,479,058
2004-12-22 00:00:00	4.6235	4.6401	4.6538	4.5582	4.5803	156,865,776
2004-12-23 00:00:00	4.6632	4.68	4.6974	4.6326	4.6687	145,125,936



Discussion :

Model Performance:

The LSTM model was able to capture the long-term trends of stock prices, thus it can be used for financial forecasting. The model was able to identify general patterns and market momentum by analyzing 20 years of Google Inc. stock data.

Feature Engineering Impact:

The addition of technical indicators such as moving averages for 100, 200, and 250 days increased the accuracy of the model since it provides useful information in terms of market trends within different time frames.

Importance of Data Preprocessing:

Data normalization through Min-Max scaling along with the train-test split ensures that the model has zero scale-related biases, making it more generalizable.

Practical Relevance:

LSTM models are highly promising for investors and analysts to make data-driven decisions. Inclusion in trading systems would improve investment strategies, particularly in long-term stock predictions.

Conclusion :

This study explored the use of Long Short-Term Memory (LSTM) networks in forecasting Google Inc. (GOOG) stock prices using 20 years of historical data, and also we can take other companies, based on stock historical data to predict. In this study, by applying moving averages and using robust preprocessing techniques such as Min-Max scaling and data splitting, it is found that LSTM models are quite good at capturing long-term price trends. However, the model faced limitations in predicting short-term fluctuations, which are heavily influenced by external market factors not accounted for in historical data.

The results indicate that LSTM networks are a promising tool for reliable exploration of market dynamics and improvements in investment strategies. External features such as sentiment analysis, macroeconomic indicators, or real-time news data might further improve the accuracy of future predictions. This study is a starting point for applying deep learning methods in financial forecasting, with the increasing importance of machine learning in modern trading and investment analysis.

REFERENCES :

1. Kim, Y. (2018). Stock Price Prediction Using LSTM Neural Networks. *Journal of Financial Engineering*.
2. Gupta, P., & Sharma, R. (2020). Stock Price Prediction Using LSTM Neural Networks. *International Journal of Machine Learning*.
3. Brownlee, J. (2017). *Deep Learning for Time Series Forecasting*. Machine Learning Mastery.
4. Zhang, Z., & Xu, Z. (2019). Time Series Forecasting for Stock Prices Using LSTM. *Journal of Machine Learning Research*. K.Senthamarai Kannan, P. Sailapath Sekar,M.Mohamed Sathik and P. Arumugam Research paper: Financial Stock Market Forecast using Data MiningTechniquesJournal: International Multi-Conference of Engineers and Computer Scientists 2010
5. Marc-André Mittermaye Research paper: Forecasting Intraday Stock Price Trends with Text Mining Techniques Journal: Hawaii International Conference on System Sciences –2004.
6. K. jae Kim, “Financial time series forecasting using support vector machines,” Neurocomputing, vol. 55, 2003.
7. Gupta, A. :“Stock market prediction using Hidden Markov Models”, IEEE Engineering and Systems (SCES), 2012 Students Conference on, 2012.
8. Lipo W., Shekhar G.: “Neural Networks and Wavelet De-Noising for Stock Trading and Prediction”, Springer, Time Series Analysis, Modeling and Applications Intelligent Systems Reference Library Volume 47, pp 229-247, 2013.
9. Suykens, J.A.K.; Vandewalle, J. "Least squares support vector machine classifiers", Neural Processing Letters, 9 (3), 293-300., 1999.
10. Guo Z., Wang H., Liu Q. : “Financial time series forecasting using LPP and SVM optimized by PSO” © Springer, Soft Computing Methodologies and Applications , December 2012.
11. Carlos A. Coello, Gary B. Lamont, David A. van Veldhuizen: “Evolutionary Algorithms for Solving Multi-Objective Problems”,Springer, 2007.
12. D. N. Wilke. “Analysis of the particle swarm optimization algorithm”, Master's Dissertation, University of Pretoria, 2005.
13. Khalil A.S.: “An Investigation into Optimization Strategies of Genetic Algorithms and Swarm Intelligence”. Artificial Life (2001).