Stock Market Prediction Using LSTM

Aakash Divakar
Electrical and Computer Engineering Dept
Virginia Tech, VA 24061
Aakash4869@vt.edu

Abstract

The finance sector has always been the epicenter of generating and controlling the flow of money. Stock market is an important part of a financial system and stock prediction analysis has always been a difficult task. Analyzing and investing have always been risky. Stock market prediction means to try and determine the future value of a stock on a stock exchange. The more detailed analysis the more accurate the prediction and so is the gain of the investor. This paper aims to identify the historical patterns to predict the trend of a stock and its price on a future date using the machine learning model Long-term short memory (LSTM). Such a model can be used to predict the prices of a share listed on any stock exchange. The LSTM model is chosen as it works well on time series data. The paper first selected the features that affect the pattern of a stock or a stock exchange and then narrowed it down to 4 features that give the best results. This study also explores using the change in price action between two intervals for a stock to identify patterns. Based on the data of the previous 30 days the trend or the price on the 31st day was predicted in the paper. The scaled data was then given input to the LSTM model with a drop layer to avoid overfitting. Finally, the result was evaluated based on the test data, and graphs were plotted to get a better understanding.

1. Introduction

The stock market, a complex and dynamic financial ecosystem, has long captivated the attention of investors, economists, and researchers alike. Stock market data is non-linear and extremely difficult to predict with many researchers and analysts proposing different methodologies to identify the patterns. There has been extensive research and work done on stock market prediction. Many computer scientists and economists have tried to design a model that can predict the price of a stock with utmost accuracy. As the nature of the stock market is extremely non-linear no one has achieved an accuracy of 100%. Even though researchers and scholars have designed models of their own or used previous models with different features and weights, no algorithm has achieved a feat of 100%. There is still research going on with the current advancements in machine learning and trying to get a high precision score or prediction rate.

2. Background

In past research work, "A Machine Learning Model for Stock Market Prediction" by Hegazy, Soliman, and Salam [1] presents a novel machine learning model combining Particle Swarm Optimization (PSO) and Least Square Support Vector Machine (LS-SVM). Addressing the unpredictability of stock markets and the limitations of traditional models, the proposed approach optimizes prediction accuracy by using PSO for parameter tuning in LS-SVM. Evaluated on thirteen financial datasets, it outperforms the ANN with the Levenberg-Marquardt algorithm, offering a significant advancement in the accuracy of financial forecasting. This model presents a breakthrough in predicting stock market trends, enhancing the tools available for financial analysis. A detailed analysis of deep learning networks was done in the research paper "Applications of Deep Learning in Stock Market Prediction: Recent Progress" by Weiwei Jiang [2]. It surveys recent works, categorizes various neural network structures, and discusses common evaluation metrics, implementation, and reproducibility issues. The paper aims to assist researchers in staying updated with rapid advancements in this field and in reproducing previous studies as baselines It explains the performance of various Neural Networks such as LSTM, GRU, StockNet, etc. Researchers investigate the precision of LSTM in predicting stock prices and examine how training epochs improve model performance by using daily opening prices of GOOGL and NKE stocks from the New York Stock Exchange, employing varying epochs for model training and testing [3]. Another study delves into the utilization of Long Short-Term Memory (LSTM) networks, a sophisticated neural network architecture, to forecast the S&P 500 index's closing price. Leveraging a meticulously selected mix of fundamental, macroeconomic, and technical indicators, this research aims to capture the multifaceted nature of stock market movements. By comparing single-layer and multilayer LSTM models, we seek to unveil a predictive approach that balances high accuracy with practical applicability, addressing the intricate patterns of market fluctuations [4]. The paper titled Stock 'Stock Price Prediction Using LSTM on Indian Share Market.' [5], explores the burgeoning field of financial market forecasting through machine learning, specifically focusing on Recurrent Neural Networks (RNN) and Long Short-Term Memory (LSTM) models. The volatility and unpredictability of financial markets present a significant challenge for traditional predictive models. Our research aims to construct an LSTM-based model to forecast future stock market values with enhanced precision. We investigate the effectiveness of LSTM in capturing complex market dynamics and examine how varying the number of training epochs influences the model's predictive accuracy. This study contributes to the evolving landscape of quantitative finance by integrating advanced machine learning techniques.

This research unlike other similar studies also focuses on the change in price between two intervals and not just the closing price of the stock for that interval. This gives the model more accurate data to work on. The main work of this paper is to achieve the best result in price and trend prediction of a stock by deciding the best features affecting the price action using LSTM. The paper demonstrates how applying various features gives poor results and leads to overfitting.

3. Dataset

The dataset will be taken from Yahoo Finance which has the entire historical data of a stock exchange. The dataset consists of the closing price, opening price, and high, and low-price during a given time interval. A ticker will be used to traverse through the dataset and get the required stock from the stock exchange it is listed on. For this study 'AAPL' ticker for Apple's stock was used.

4. Methodology

Figure 1: Schematic diagram of the proposed research

4.1 Finalizing Dataset

The data from Yahoo Finance [6] was kept limited to Closing Price, High, Low, and Opening Price, and the rest of all columns were removed. The data was sliced and kept from 1st January 1991 to the present, this was done as the market could have shifted fundamentally and historical data too far back would not be useful. Further, three more columns were added to this dataset which are 'Target,' 'Tomorrow,' 'Target Class' and 'US 10-year bond yield'. The Target column consists of the difference between the Closing price of the current interval and the previous interval. The Tomorrow column is the closing price of the next day, this was done by shifting the Close column by -1. The Target Class dictates the trend of the stock, if the Target Class is 1 the trend is upward or the price of the stock has increased, and 0 for downward trend. In this research, the time interval was kept as one day.

	Open	High	Low	Close	Target	US10yr	Tomorrow	TargetClass
Date								
1991-01-02 00:00:00-05:00	0.306061	0.315010	0.300692	0.311431	0.005370	7.970	0.307852	0
1991-01-03 00:00:00-05:00	0.311431	0.316800	0.307852	0.307852	-0.003579	7.930	0.309641	1
1991-01-04 00:00:00-05:00	0.307852	0.316800	0.307852	0.309641	0.001790	8.020	0.309641	0
1991-01-07 00:00:00-05:00	0.307852	0.323960	0.307852	0.309641	0.001790	8.130	0.309641	0
1991-01-08 00:00:00-05:00	0.313221	0.314116	0.304271	0.309641	-0.003579	8.160	0.323960	1
...	...	...	...	...	...	...	...	...
2023-11-13 00:00:00-05:00	185.820007	186.029999	184.210007	184.800003	-1.020004	4.632	187.440002	1
2023-11-14 00:00:00-05:00	187.699997	188.110001	186.300003	187.440002	-0.259995	4.441	188.009995	1
2023-11-15 00:00:00-05:00	187.850006	189.500000	187.779999	188.009995	0.159988	4.535	189.710007	1
2023-11-16 00:00:00-05:00	189.570007	190.960007	188.649994	189.710007	0.139999	4.445	189.690002	0
2023-11-17 00:00:00-05:00	190.250000	190.380005	188.574997	189.690002	-0.559998	4.441	NaN	0

8284 rows × 8 columns

Figure 2: Dataset after cleaning

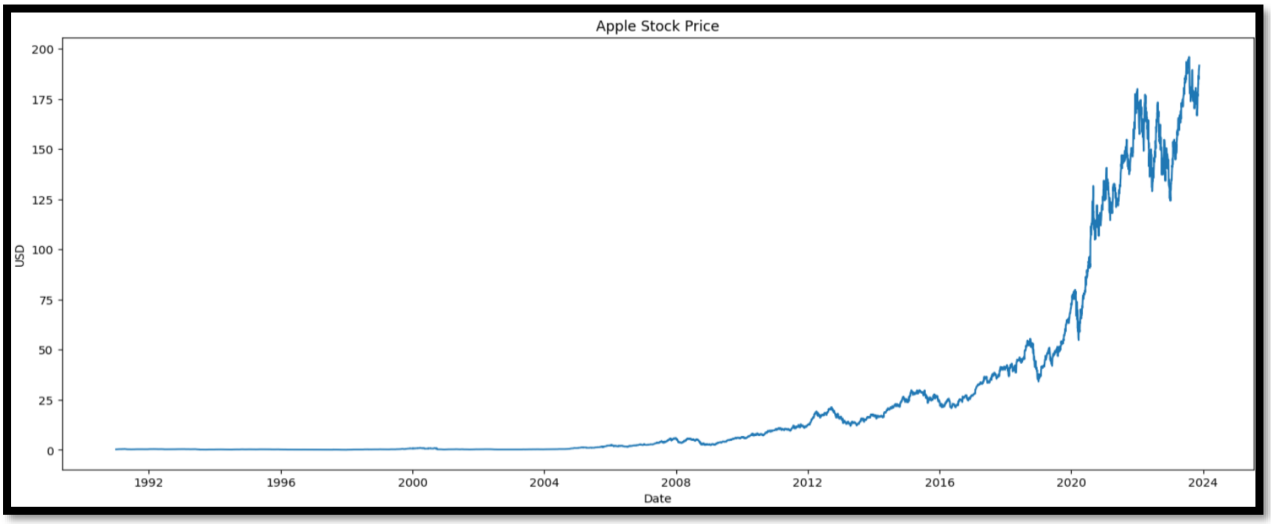


Figure 3: Apple's Stock line chart

4.2 Indicators

The stock market consists of numerous indicators that help investors in deciding the entry point and the exit point of a stock. The most commonly used and prominent indicators were selected and added to the dataset to further help in predicting the target price. The indicator 'EMA' (exponential moving average) is a technical indicator commonly used on price charts to monitor the price movements of investments such as stocks or commodities over a period of time. Unlike a simple moving average (SMA), the EMA is a type of weighted moving average (WMA) that places greater emphasis on recent price data. Similar to the SMA, the EMA helps in identifying price trends over time, and it's possible to observe multiple EMAs simultaneously, often visualized as moving average ribbons. The 'MACD' (Moving average convergence/divergence) is a momentum oscillator that calculates the difference between the 12-day exponential moving average (EMA) and the 26-day EMA. Unlike a simple moving average, EMA gives more weight to recent data points. Traders use the MACD to assess trends, direction, momentum, and potential reversals in stock prices. For instance, if the MACD curve crosses above the zero line, it suggests a buying opportunity, while a crossover below the zero line indicates a selling opportunity. Research findings suggest that the MACD indicator holds significant potential for making well-informed investment decisions. The 'MA7,' (7- day Moving Average) and 'MA21' (21- day Moving Average) were also added. The MA7 provides a short-term view of price trends, responding quickly to recent market developments. Traders often use it to gauge immediate momentum and make short-term trading decisions. On the other hand, the MA21 offers a more extended perspective, smoothing out short-term fluctuations to reveal broader market trends. It helps investors identify longer-term patterns and potential support or resistance levels. Together, these moving averages provide valuable insights into market dynamics, assisting traders and investors in making informed decisions.

Date	Open	High	Low	Close	Target	US10yr	Tomorrow	TargetClass	EMA	MA7	MA21	MACD
1991-01-02 00:00:00-05:00	0.306061	0.315010	0.300692	0.311431	0.005370	7.970	0.307852	0	0.311431	NaN	NaN	-0.003579
1991-01-03 00:00:00-05:00	0.311431	0.316800	0.307852	0.307852	-0.003579	7.930	0.309641	1	0.308746	NaN	NaN	-0.005713
1991-01-04 00:00:00-05:00	0.307852	0.316800	0.307852	0.309641	0.001790	8.020	0.309641	0	0.309366	NaN	NaN	-0.005922
1991-01-07 00:00:00-05:00	0.307852	0.323960	0.307852	0.309641	0.001790	8.130	0.309641	0	0.309552	NaN	NaN	-0.007208
1991-01-08 00:00:00-05:00	0.313221	0.314116	0.304271	0.309641	-0.003579	8.160	0.323960	1	0.309612	NaN	NaN	-0.006785
...	...	...	...	...	...	...	...	...	...	...	...	...
2023-11-13 00:00:00-05:00	185.820007	186.029999	184.210007	184.800003	-1.020004	4.632	187.440002	1	184.853408	181.858843	175.958140	-3.063395
2023-11-14 00:00:00-05:00	187.699997	188.110001	186.300003	187.440002	-0.259995	4.441	188.009995	1	186.577804	183.433476	176.384575	-3.487114
2023-11-15 00:00:00-05:00	187.850006	189.500000	187.779999	188.009995	0.159988	4.535	189.710007	1	187.532598	184.721449	176.912817	-3.958608
2023-11-16 00:00:00-05:00	189.570007	190.960007	188.649994	189.710007	0.139999	4.445	189.690002	0	188.984204	185.882767	177.584311	-4.398570
2023-11-17 00:00:00-05:00	190.250000	190.380005	188.574997	189.690002	-0.559998	4.441	NaN	0	189.454736	186.888572	178.272922	-4.619638

Figure 4: Final Dataset

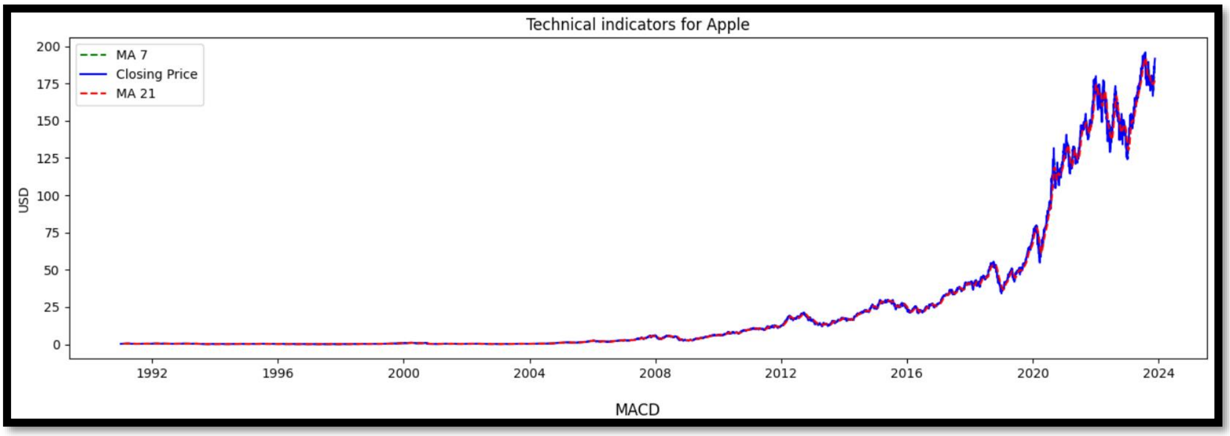


Figure 5: Plot of MA7 and MA21

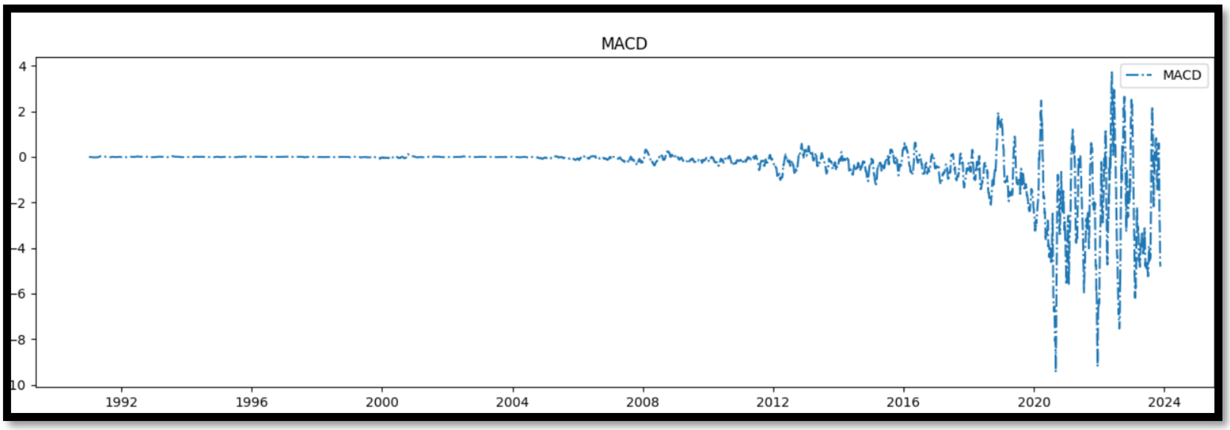


Figure 6: Plot of MACD

4.3 Data Normalization

After cleaning and finalizing historical data, it needs to be normalized as normalization helps to remove the impact of scale and put all features on the same scale. This is achieved using the inbuilt function MinMaxScaler [7]. This function scales and translates each feature independently such that it is in the given range on the training set, e.g., between zero and one

```
X_std = (X - X.min(axis=0)) / (X.max(axis=0) - X.min(axis=0))
X_scaled = X_std * (max - min) + min
```

Figure 7: MinMaxScalar equation

4.4 Train and Test Dataset

Training data was kept 80% of the total data and the remaining 20% was used for testing and validation. This research experimented with 7 days, 30 days, and 60 days of previous data for prediction of the next day and found 30 days of data to predict the 31^{st} day gave prominent results and accordingly the scaled data was augmented for training purposes for all the required features.

4.5 LSTM Model

A baseline Recurrent Neural Network (RNN) has loops, and information to persist. In Figure 7, a chunk of neural network, A, looks at some input x_t and outputs a value h_t. A loop allows information to be passed from one step of the network to the next. Using this single cell, a chain of networks can be created which passes on the information to the other.

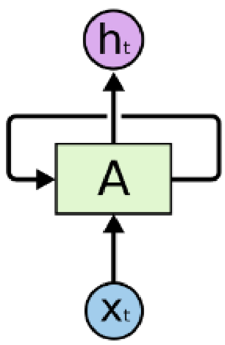
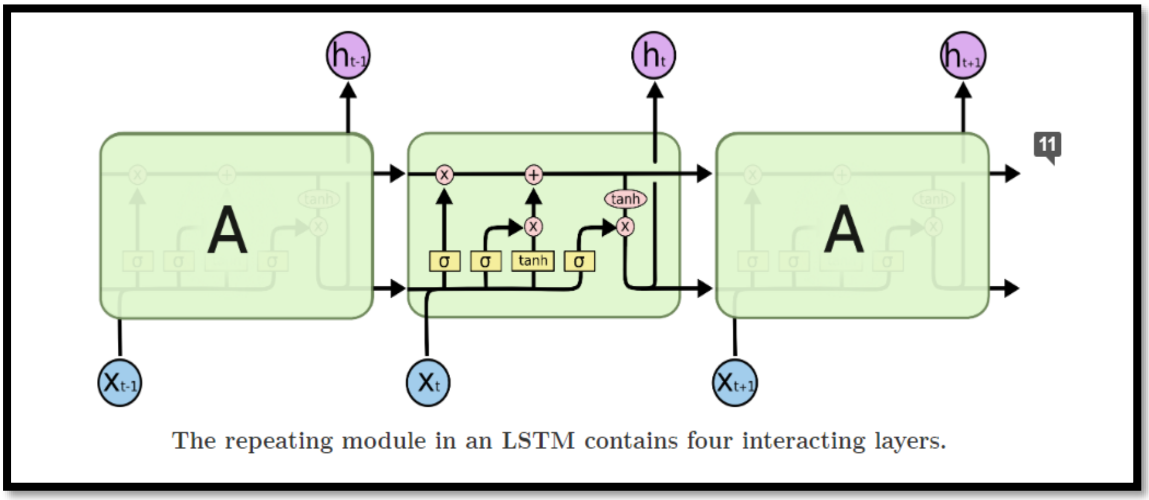


Figure 8: RNN cell

The machine learning algorithm used for stock market prediction in this project is Long short-term memory (LSTM). LSTM is an advanced version of RNN, the RNNs have loops allowing them to use past information to predict the future but RNNs can retrieve only recent information and they also suffer from vanishing gradient problems. The LSTMs as the name suggests can remember information over a long period also, they work well with time series data. LSTMs have also outperformed Deep neural networks (DNNs) as the DNNs provide modeling for a fixed-sized sliding window where the network does not interdepend on the previous time steps which would not provide good modeling for the stock data and hence the reason for selection of the algorithm. [8]



The repeating module in an LSTM contains four interacting layers.

Figure 9: LSTM Cell

The LSTMs have the ability to add or remove information as needed which they achieve using a gate. There are three gates in LSTM which are basically composed of sigmoid function and can control what to let through and what to stop from getting through. A value of 1 means passes everything through and the opposite for a value of 0. The first and a important layer of LSTM is the forget gate layer which dictates what to let through and what to forget.

$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] \ + \ b_f\right)$$

The next gate is used to decide what new information we have to store in the cell state. This is divided into two parts, firstly the sigmoid layer decides which values to update then the tanh layer creates a vector that could be added to the state afterwards both will be combined to create an update in the state

$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] \ + \ b_i\right)$$
$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] \ + \ b_C)$$

Now, we update the old cell state to the new state. We multiply the old state with the forgetting things that we decided to forget then add the new information to be added.

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$

For the last step, we decide what should be the output. We first go through a sigmoid layer which governs what parts to output then we go through tanh. This is done to make sure the values are between +1 and -1.

$$o_t = \sigma\left(W_o\ [h_{t-1}, x_t]\ +\ b_o\right)$$

$$h_t = o_t * \tanh\left(C_t\right)$$

4.6 Feature Selection

The stock market analysis involves many indicators and a ton of factors that determine the price action. This research considered 10 features that impact the price of the stock the most and experimented with the most important ones and finally selects the features that provide the closest prediction.

1) Close price – The closing price of the stock is important as it follows a pattern and helps in determining the mood of the market for the next day.

2) Open price – The opening price of stock helps in the same way as the closing price of stock.

3) High price – The highest price the stock has climbed in a period of time gives information about the uptrend

4) Low price – The lowest price of stock helps in determining the support of a stock in a time frame.

5) United States 10-year bond yield – The US 10-year bond yield is viewed as a safe investment as it is backed by the US government. The bond yield and stock market usually go opposite ways. This is because when people find higher returns elsewhere, they think they need not play safe.

6) The MACD indicator – The Moving average convergence/divergence indicator is calculated using two EMAs (Exponential moving average). A MACD crossover helps in determining if the market will move up or down.

7) The RSI indicator. The Relative Strength Index helps in evaluating the price of a stock if it is oversold or overbought.

8) Moving Average: Moving Average of 7 days and 21 days was taken into consideration.

9) Exponential Moving Average: EMA was calculated and used as a essential pattern identifier for the research.

10) Interest Rate – The Federal Reserve of the US manages the monetary policies and sets interest rates which ultimately leads to cash flow in the economy. This increases or decreases the inflation in the market.

4.7 Training the model

The training dataset after normalizing and splitting into 80% of the total data was fed to the LSTM model. The data is given to the input layer which takes the input shape as input dimension and the number of features. The input layer was kept Bidirectional as it can improve model performance on sequence classification problems. Bidirectional LSTMs train two LSTMs on the input sequence rather than one when all timesteps of the input sequence are accessible. the first on the input sequence exactly as it is, and the second on an input sequence duplicate that has been inverted. This can provide the network with more context and enable it to learn about the issue more quickly and thoroughly. Two dense layers were added between the input layer and the output layer. After each layer before the output layer, a dropout layer was introduced to reduce overfitting. The dropout rate was kept at 0.1. The Dropout layer randomly sets input units to 0 with a frequency of rate at each step during training time, which helps prevent overfitting. Inputs not set to 0 are scaled up by 1/(1 - rate) such that the sum of all inputs is unchanged. [9]

$$h' = \begin{cases} 0 & \text{with probability } p \\ \frac{h}{1-p} & \text{otherwise} \end{cases}$$

Figure 10: Dropout equation

The model was compiled with the optimizer algorithm and its associated hyperparameters. For this research ‘Adam’ optimizer was used with a learning rate of 0.01. Clipnorm was also introduced which helps control the gradient values during training by clipping them if they exceed this threshold to 1. Adaptive moment estimation, or "Adam" optimizer, is a term for an iterative optimization process that reduces the loss function when neural networks are being trained. Adam may be thought of as a stochastic gradient descent with momentum combined with RMS prop.[10] Where w is model weights, b is for bias, and eta is the step size which can depend on iteration. Then further I trained the model using a batch size of 32 with 25 epochs.

$$\boxed{\begin{aligned} W_t &= W_{t-1} - \eta' V_{dw} \\ b_t &= b_{t-1} - \eta' V_{db} \end{aligned}} \qquad \boxed{\eta' = \frac{\eta}{\sqrt{S_{dw} + \epsilon}}}$$

Figure 11: Adam Algorithm

5. Evaluation and Results

The model was evaluated with the combination of different features according to their importance and effect on the stock market. The number of features fed to the model was varied and experimented with 10 to 50 epochs. The feature ‘Close’ i.e., the closing price was kept in all the combinations as it is the essence of prediction. The study finds that using more epochs led to overfitting and even with the use of dropout layers the result did not change. The use of all 10 features also did not give prominent results therefore the study was continued with a smaller number of features. The model also suffered from overfitting after 20-25 epochs and hence the final epochs were set to 25 after careful research.

I started experimenting with just one feature and eventually increased it while evaluating different combinations of features with increasing importance. The first test was carried out with the ‘Close’ feature and the results were not satisfying. In Figure 10, the orange line shows the predicted chart and the green line shows the actual chart of Nifty 50, a stock exchange Index in India. The result was obtained by using only the closing price of the exchange as a feature.

Figure 12: Nifty 50 prediction

Further, I tried using the ‘Close,’ ‘US 10-Year yield’ and the ‘EMA’ indicator as features. The US 10-year bond yield was chosen as the stock market reacts to movement in the bond yield over the long term and thus could lead to accurate results. EMA gives the average movement of the stock over a period and these two features could bring a better result. The result obtained was far better than just the use of one feature. Similarly, I tried adding the ‘MA7’ and the ‘Target’ price as features which significantly increased the efficiency.

Comparing various combinations of features and several features give different result but the model performed poorly after adding 5 or more than 5 features for any combination. This was observed throughout the experiment and hence the model was only fed 4 features to give accurate results. The model gave the best result when ‘Close’, ‘US 10-year bond yield,’ ‘MA21’and ‘Target’ price was fed to it. The performance observed in the Apple’s stock price is shown in the figure 11. The prediction line follows very closely with the test or the actual line chart of the Apple stock.

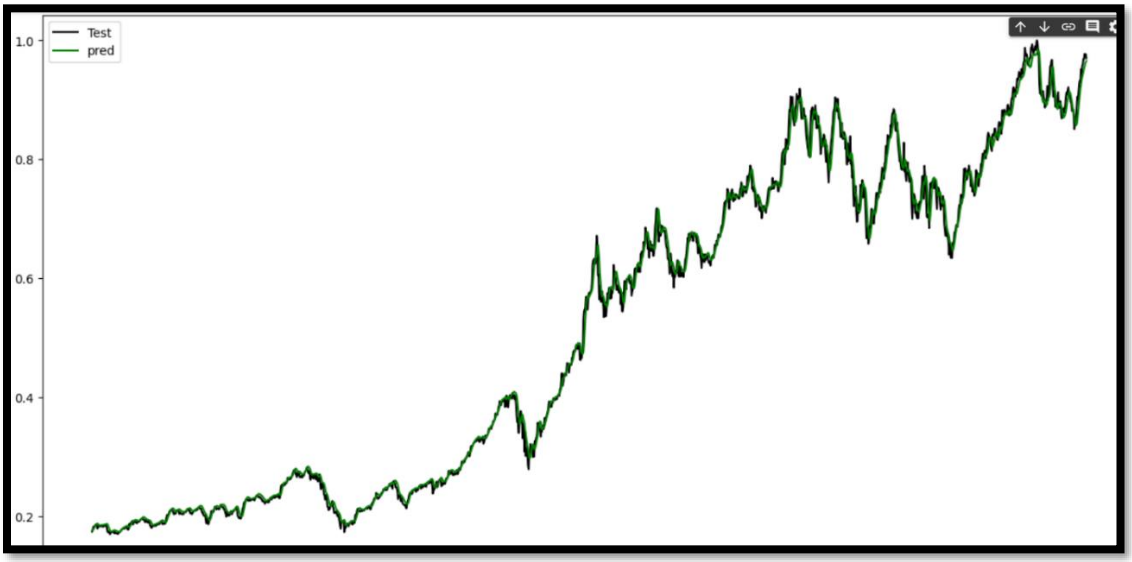


Figure 13: Apple's Stock price prediction

6. Conclusion and Future Work

The results obtained are promising and will definitely help an investor before taking a trade. Even though the predicted price is not precise, it gives the investor a sense of knowledge of the historical pattern followed by the stock. The problem with the stock market is that it has an extremely non-linear nature and any event can change its course radically. There is a scope for better prediction by analyzing the change in stock price or change I pattern instead of the actual price/pattern to detect a future possibility of a turn in the market. Instead of predicting the exact price, predicting a change in stock price by a certain amount will help investors decide their trade. The column ‘Target Class’ shows if the stock price went up or down and predicting this is also helpful to make a decision for entry or exit in a stock.

7. References

[1] Hegazy, Osman & Soliman, Omar S. & Abdul Salam, Mustafa. (2013). A Machine Learning Model for Stock Market Prediction. International Journal of Computer Science and Telecommunications. 4. 17-23.

[2] Weiwei Jiang, Applications of deep learning in stock market prediction: Recent progress, Expert Systems with Applications, Volume 184, 2021, 115537, ISSN 0957-

4174.
[3] Moghar, A., & Hamiche, M. (2020). Stock Market Prediction Using LSTM Recurrent Neural Network. In Proceedings of the International Workshop on Statistical Methods and Artificial Intelligence (IWSMAI 2020) (pp. [page numbers if available]). April 6-9, Warsaw, Poland
[4] 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, 100320, ISSN 2666-8270.
[5] Ghosh, Achyut & Bose, Soumik & Maji, Giridhar & Debnath, Narayan & Sen, Soumya. (2019). Stock Price Prediction Using LSTM on Indian Share Market. 10.29007/qgcz
[6] Yahoo Finance. (2023). Apple Inc. (AAPL) Stock Historical Prices & Data. Retrieved from https://finance.yahoo.com/quote/AAPL/history
[7] scikit-learn developers. (2023). sklearn.preprocessing.MinMaxScaler. Retrieved from https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.MinMaxScaler.html
[8] Christopher, Olah. "Understanding LSTM Networks," August 2017.
[9] TensorFlow. (2023). tf.keras.layers.Dropout. Retrieved from https://www.tensorflow.org/api_docs/python/tf/keras/layers/Dropout
[10] Analytics Vidhya. (2023). What is Adam Optimizer? Retrieved from https://www.analyticsvidhya.com/blog/2023/09/what-is-adam-optimizer/
[11] Pramod, B. S., and Mallikarjuna Shastry Pm. "Stock price prediction using LSTM." *Test Engineering and Management* 83 (2020): 5246-5251.
[12] Pawar, Kriti & Jalem, Raj & Tiwari, Vivek. (2019). Stock Market Price Prediction Using LSTM RNN: Proceedings of ICETEAS 2018. 10.1007/978-981-13-2285-3_58.
[13] Thushan, Ganegedara. "Stock Market Predictions with LSTM in Python," December 2019.
[14] Pramod, & Pm, Mallikarjuna. (2021). Stock Price Prediction Using LSTM. Test Engineering and Management. 83. 5246-5251.
[15] Yadav, A., Jha, C. K., & Sharan, A. (2019). Optimizing LSTM for time series prediction in Indian stock market. In Proceedings of the International Conference on Computational Intelligence and Data Science (ICCIDS 2019)