STOCK MARKET PREDICTION USING MACHINE LEARNING

Saroj Doranga[1], Rajeev Nepal[2], Pratigya Timsina[3]

[1]Saroj Doranga
Email: 9846656375saroj@gmail.com
Department of Electronics and Computer Engineering, Thapathali Campus, Institute of Engineering, Kathmandu, Nepal

[2]Rajeev Nepal
Email: nepalrajeev11@gmail.com
Department of Community Medicine, Universal College of Medical Sciences and Teaching Hospital (UCMS-TH)

[3]Pratigya Timsina
Email: pratigyatimsina324@gamil.com
MBS Scholar, Prithvi Narayan Campus, Tribhuvan University, Pokhara, Nepal

ABSTRACT

The act of buying and selling stocks is a crucial activity in the realm of finance. The complexity and volatility of the stock market have made predicting its behavior a difficult task. Although traditional methods of technical and fundamental analysis exist, they have limitations in terms of accuracy and speed. Recently, there has been an increasing interest in leveraging machine learning to improve the prediction of stock market trends. Machine learning algorithms have proven to be more efficient in predicting the stock market compared to traditional methods. However, predicting the stock market is a complex task that requires identifying various attributes to train the algorithm. Our review of relevant literature has determined that the most effective machine learning tools for this research are Artificial Neural Network (ANN), Support Vector Machine (SVM), and Genetic Algorithms (GA), each with its own distinct strengths and limitations. We collected data from the inception of Netflix and used Long Short-Term Memory (LSTM) Neural Network-based machine learning models to analyze and predict its stock price. Additionally, the Recurrent Neural Network (RNN) is useful in preserving time-series features to enhance profitability.

Key Words Genetic Algorithms, Long Short Term Memory (LSTM), Recurrent Neural Network (RNN), Stock Prediction

Introduction

Predicting the long-term value of stocks and other financial instruments traded on financial exchanges is a crucial aspect of the stock market. Investors, traders, and financial analysts face significant challenges in predicting stock market trends due to its complexity and volatility. Although traditional methods, such as technical and fundamental analysis, have been used to predict stock prices, they are limited in terms of accuracy and speed. To overcome these limitations, researchers have turned to machine learning algorithms, a subset of artificial intelligence that uses statistical techniques to enable computers to learn and improve their performance (Shah et al., 2019). Most stockbrokers use technical and fundamental analysis, as well as statistical analysis, to predict stock behavior. The value of stocks is an important indicator of an individual's wealth and popularity (Bhandari et al., 2022). The stock market provides a platform for buying and selling stocks, with the buyers and sellers receiving their money within a few days, depending on the country's regulations. As finance becomes increasingly important, more people are taking an interest in the stock market (Islam et al., 2021).

As a result, researching stock market prediction has become more critical. Machine learning has demonstrated significant success in various fields, including natural language processing, image recognition, and medical diagnosis (Kim et al., 2012). In the financial industry, it has become increasingly popular for predicting stock market trends. This research aims to investigate the effectiveness of machine learning algorithms, including neural networks, support vector machines, decision trees, and random forests, in predicting stock market trends (Naik & Mohan, 2019). Machine learning algorithms provide various predictive techniques that can be useful in avoiding losses. Various sectors use different traditional techniques, but the growing field of machine learning can improve their performance. Stock prediction involves two types of analysis: fundamental and technical. Fundamental analysis includes factors such as growth and past performance, while technical analysis considers volume, prices, and time-based features like open and close prices (Moghar & Hamiche, 2020).

In this project, the LSTM algorithm of machine learning was used to increase the accuracy of stock price prediction (Pramod & Pm, M. S, 2020). During the literature review, it was found that decision trees, SVM, and other machine learning algorithms are also useful. Different algorithms have unique findings and limitations. LSTM-based neural network models can analyze and predict stock prices better than other methods (Sunny et al., 2020). The Recurrent Neural Network (RNN) is particularly useful as it considers past values, preserving the time-series features for improved profitability from stock (Ortega & Khashanah, 2014). The primary goal of this project was to predict the future price of a particular stock using machine learning algorithms (Bhalke et al., 2022).

The research focuses on predicting stock prices of companies listed on the stock market using various datasets, such as historical stock prices, financial ratios, and news sentiment data. The performance of each algorithm is evaluated using different evaluation metrics, such as accuracy, precision, recall, and F1-score. The results of this study will provide valuable insights into the potential of machine learning in predicting stock market trends and its potential to benefit investors, traders, and financial analysts.

Literature Review

Various methodologies are employed to model market movement and volatility, including the use of feathers and data from foreign stock like Netflix. In this research, data will be collected from Yahoo Finance. However, similar projects have been conducted in the field of machine learning using different algorithms to predict stock trends. The use of machine learning techniques for stock market prediction has been an active research area in recent years (Kai & Wenhua, 1997). Several studies have been conducted using various machine learning algorithms, including artificial neural networks, support vector machines, decision trees, and deep learning methods like recurrent neural networks and long short-term memory (Islam et al., 2021).

In a study by Chen et al. (2015), LSTM-based method was proposed for stock return prediction using data from the Chinese stock market. The authors achieved promising results with their model, outperforming other traditional machine learning methods.

Another study by Roondiwala et al. (2017) used LSTM to predict stock prices. The authors used data from the Bombay Stock Exchange and reported good accuracy in their predictions.

In a comparative study by Karmiani et al. (2019), the authors compared the performance of several predictive algorithms, including backpropagation, SVM, LSTM, and Kalman filter for stock market prediction. The authors concluded that LSTM outperformed other algorithms in terms of accuracy.

Althelaya et al. (2018) evaluated the performance of bidirectional LSTM for short and long-term stock market prediction. The authors used data from the Saudi stock market and reported improved accuracy in their predictions compared to traditional machine learning methods.

Furthermore, study by Biswas et al. (2021) proposed a logical strategy using deep learning for stock market prediction. The authors used a combination of deep learning algorithms and technical indicators for their model and reported good accuracy in their predictions.

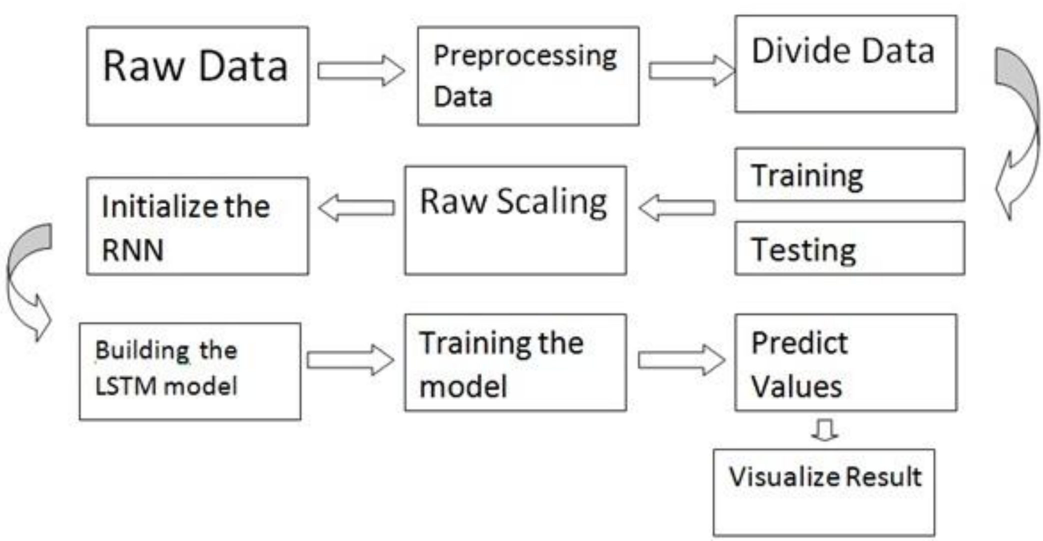
Overall, these studies suggest that machine learning techniques, particularly deep learning methods, can be effective for stock market prediction. However, the choice of algorithm and data used can significantly impact the accuracy of the predictions (Moghar & Hamiche, 2020). For instance, Dr. J. Dhilipan, D.B. Shanmugam, and Imran Quraishi used decision trees and LSTM for trend analysis, achieving high prediction efficiency using LSTM. Similarly, Osama Assaf conducted research on Multivariate LSTM for Stock Market Volatility Prediction to improve volatility forecasting accuracy and build more profitable trading portfolios and options pricing models. Nagaraj Naik and Biju R Mohan analyzed Optimal Feature Selection of Technical Indicator and Stock Prediction using Machine Learning Technique and found that feature selection techniques are useful in identifying relevant technical indicators for stock prediction.

Methodology

Below figure illustrates the overall system block diagram representing all the tasks performed. This block diagram illustrates the overall flow of your stock market prediction system using machine learning. Raw data is first collected and then preprocessed to remove missing values, outliers, and noise (Biswas et al., 2021).The data is then divided into training and testing datasets. Raw scaling is applied to normalize the data. The RNN is initialized, and the LSTM model is built. The model is trained using the training dataset. The trained model is then used to predict stock market trends using the testing dataset (Sunny et al., 2020). Finally, the predicted results are visualized to analyze the performance of the model.

Fig 1

System Block Diagram



Data Collection and Preparation. Data for the company Netflix (NFLX) was collected from Yahoo Finance. The data includes stock information such as High, Low, Open, Close, Adjusted Close, Earnings per Share (EPS), and Volume. For training and testing the model, the Close column was selected. The data was divided into training and testing sets, with the training set being used to train the model and the testing set being used to evaluate the accuracy of the model(Roondiwala et al., 2018). The data was divided into an 80:20 ratio, with 80% used for training and 20% used for testing. To normalize the data and make it easier for the algorithm to learn patterns, the data was scaled between a range of 0 and 1.

Once the data was scaled, a training dataset was created in batches for input and output. The next step involved building the LSTM model, with the number of epochs representing the number of passes of the entire training dataset the model completes.

Loading the Stock Prices Dataset

To begin, import the CSV file was used as a DataFrame with Pandas. As the data was sorted by date, we indexed the DataFrame by date column as well. By plotting the High and Low values of Netflix stock from the beginning of trading, we observed the chart depicted in figure 2 below.

Fig 2

Stock High and Low Values

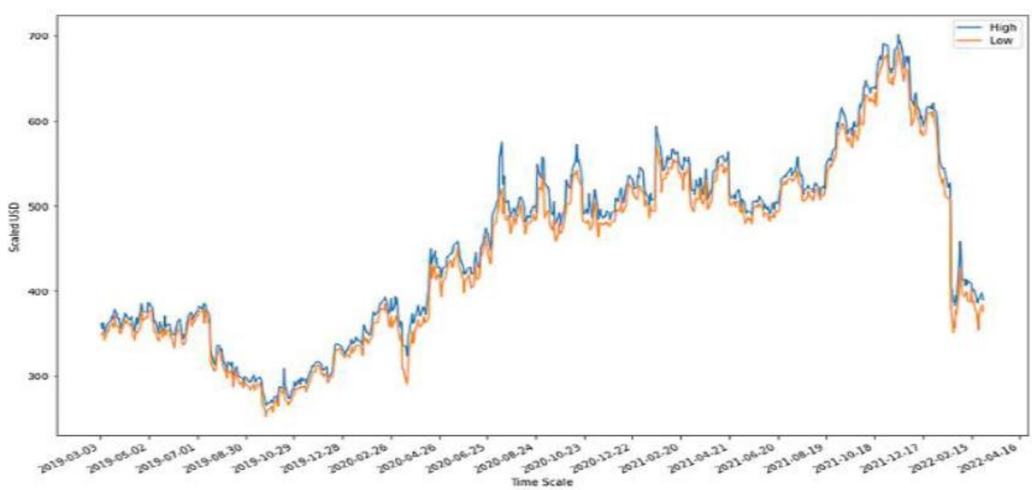
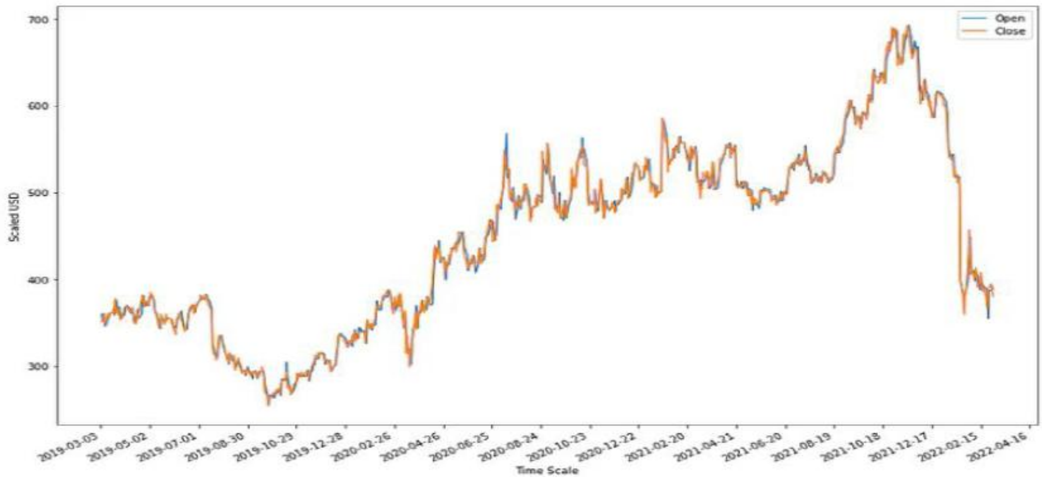


Fig 3

Stock Open and Close value



Likewise, plotting the Open and Close values for each day of stock trading provides similar observations, as demonstrated in Figure 3 above.

Recurrent Neural Network (RNN)

The LSTM is a type of Recurrent Neural Network (RNN) that utilizes information from previous records to predict future outcomes. RNNs were used to store information from previous inputs in their memory when dealing with large datasets. The unrolled and rolled RNN architectures are illustrated in below figures (4) and (5) respectively.

Fig 4

An unrolled RNN

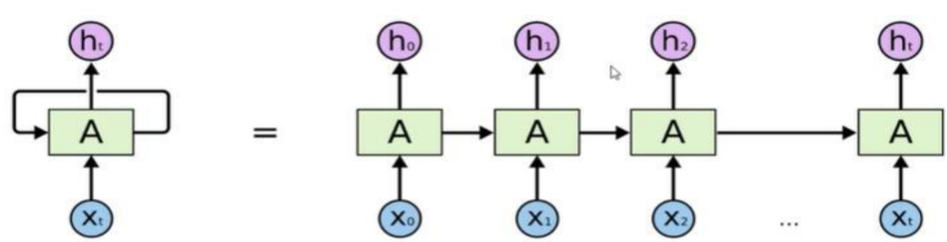
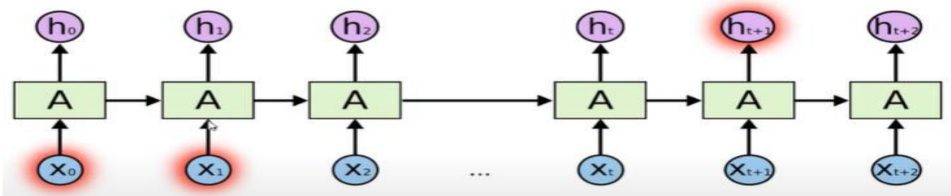


Fig5

Rolled RNN



The first output was based on the first input, the second output on the first and second inputs, and the third output on the first, second, and third inputs. RNNs can be thought of as multiple copies of the same neural network, with each copy passing a message to its successor (Althelaya et al., 2018). However, RNNs suffer from the issue of being long-term dependent, making them unsuitable for processing large sets of RNN data.

Vanished Gradient. Neural networks rely on Back propagation to adjust their weights based on the difference between their output and the actual output. When this error is small, it is multiplied by the learning rate and propagated back through the network's layers, becoming almost zero by the time it reaches the last cell (Staudemeyer et al., 2019).This phenomenon is referred to as vanishing gradient.

Exploding Gradient. Exploding gradient occurs when the gradient of the loss function becomes too large during training, causing the neural network weights to update by a large amount. Due to this, the model may become unstable and provide incorrect predictions (Hochreiter & Schmidhuber, 1997). This issue can arise if the learning rate is set too high or if the weights are initialized with very large values. To avoid this problem, several techniques like gradient clipping or weight regularization were implemented. If the algorithm gives too much significance to the weights, it can cause the values to explode as it progresses towards the final cell.

Long Short Term Memory (LSTM)

LSTM is a variant of RNNs that can handle long-term dependencies. Unlike traditional RNNs, LSTM is specifically designed to tackle long-term dependency issues and retain information for a prolonged duration. LSTM utilizes a supervised learning approach and can process entire sequences of datasets, as illustrated in Figure 6. The LSTM cell in Figure 4 is updated through controlling gates that regulate the previous memory and input. This is how LSTM handles the problems of vanishing and exploding gradients. Figure 7 provides a clear depiction of how input, output, and forget gates operate, and how the cell retains values while the gates regulate the flow of information.

Fig 6

LSTM

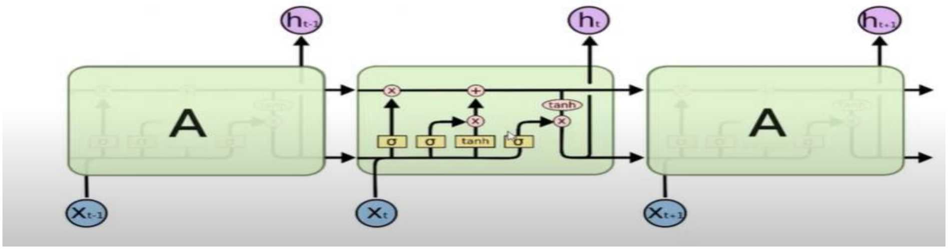
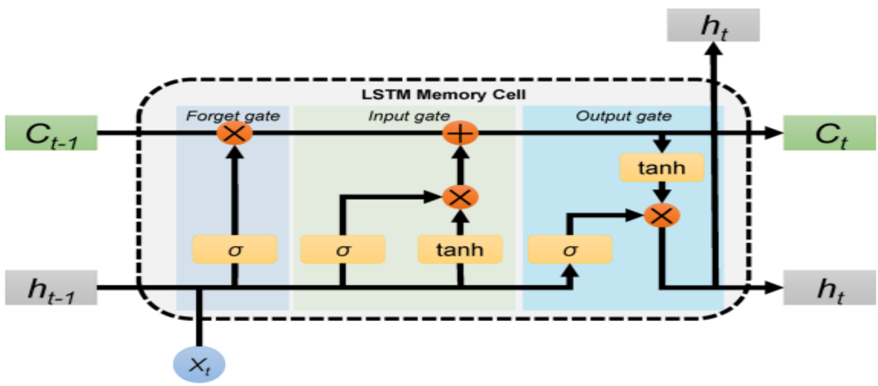


Fig 7

Memory Cell



Building the LSTM model. In order to construct the LSTM model, we imported the necessary packages and libraries, including the Keras library, which is a high-level API for developing and training models using Tensorflow. Therefore, we imported several libraries. The goal was to use these tools to construct our model.

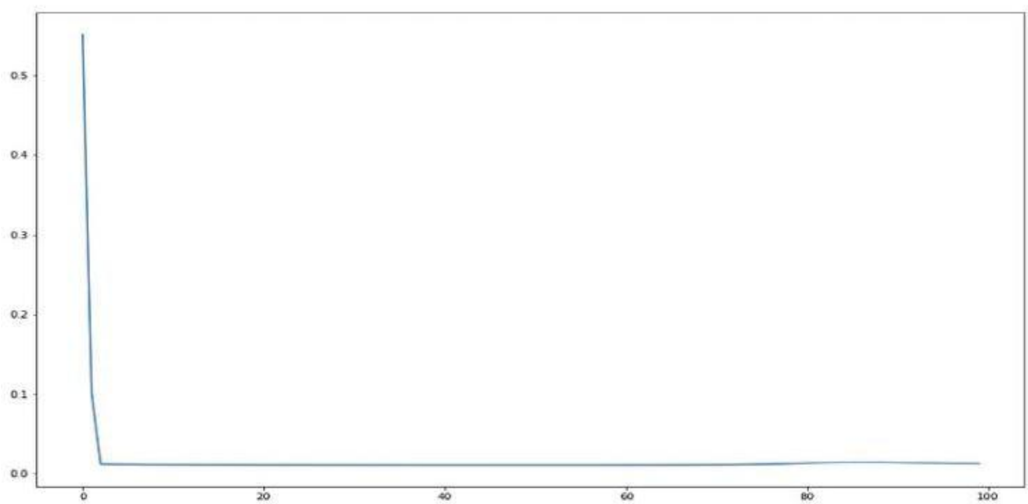
Sequential. For the creation of a linear stack of layers, we adopted sequential model in Keras. Layers were, then, added to the model one by one, in sequence.

Dense. This layer is used in neural networks to connect all the neurons from one layer to the next layer. It is the most commonly used layer in deep learning models. The Dense layer multiplies the input by a weight matrix and then adds a bias vector. The trainable parameters of the layer are updated during back propagation to optimize the model.

Next, we initialized the RNN for our time series problem using a regression model. We first read in the sequential data and assign it to a regressor. The most important step is training the model, where we fed the collected data to the model and trained it for prediction. Our LSTM model consisted of a sequential input layer followed by three LSTM layers and a dense layer, with dropout used as a regularization technique to reduce overfitting. After this, we compiled the model using an optimizer which changed the attributes of the neural network such as weights and learning rate to reduce losses. We then trained the model, with an epoch indicating how many times the algorithm has completed. To evaluate the model, we created testing data from scaled values, converted it to a NumPy array and reshape it, and used the root mean squared error (RMSE) to measure accuracy. A lower RMSE indicated better predictions. However, due to the simplicity of the model and the data, we observed that the loss reduction stagnates after only a few epochs, as shown in Figure (8) when plotting the training loss against the number of epochs, indicating that the LSTM does not learn much after 10-20 epochs.

Fig 8

Training loss against the number of Epochs



Working of LSTM. An LSTM is a unique type of network that includes three gates within its structure: the input gate, forgetting gate, and output gate, all located in the LSTM unit. The LSTM selectively processes information using pre-defined rules upon entry. Only information that adheres to the algorithm is retained, while information that doesn't comply is erased through the forgetting gate. The following steps outline the process:

Stage 1: Raw Data: During this phase, historical stock data is gathered from the internet and utilized for forecasting future stock prices.

Stage 2: Data Preprocessing: The pre-processing stage involves:

a) Data Discretization: Although it is a component of data reduction, this process holds significant importance, particularly when dealing with numerical data.

b) Data transformation: Normalizations.

c) Data cleaning: Fill in missing values.

d) Data integration: The next step involves integrating data files. Once the dataset has been transformed into a clean format, it is partitioned into training and testing sets to facilitate evaluation. The more recent values are used for the training set, whereas the testing data typically comprises 5-10 percent of the overall dataset.

Stage 3: Training Neural Network: During this phase, the neural network is fed with data and trained by assigning random biases and weights for prediction. Our LSTM model consists of a sequential input layer, 2 LSTM layers, and 4 dense layers that employ ReLU activation. Finally, a dense output layer is included.

Stage 4: Output Generation: During this step, the output generated by the output layer is compared to the target value from the dataset. The back propagation algorithm is then used to minimize the error or difference between the target and the obtained output value by adjusting the weights and biases of the network.

Data Pre-processing. To use the dataset with the model, it must first undergo data preprocessing as it is typically not in the correct format. In our model, the following steps were taken:

- Null rows were dropped
- The data types of the columns/features were modified
- The data was sorted by date
- Older data was removed

Handling Categorical Data. In the categorical data part, certain features were identified as categorical and couldn't be directly used for training the model. To handle this, Label Encoding and Frequency Encoding methods were applied. Label Encoding creates a new column for a feature and assigns a numerical value to each category based on their order of appearance. This was accomplished using the inbuilt preprocessing feature of the Sklearn library in Python, with default parameters. Any missing values in a feature were encoded as NaN values, which were later dealt with. Frequency Encoding, on the other hand, replaces categorical values with their frequency of occurrence. Python's inbuilt library was used for this method, with a parameter set to encode missing values as NaN values.

Imputing Missing Values. Variables that do not directly impact the model and are absent can be dropped. If the number of missing values in a feature is greater than 70% of the total instances in a dataset, that feature is dropped completely during preprocessing. The decision to drop such features was based on the belief that the effort required to impute the missing values would be more significant than their role in predicting outcomes. Additionally, List-wise Deletion was used to remove instances with missing values in the target feature that did not impact stock prediction. For remaining features with missing values, Multivariate Imputation by Chained Equations (MICE) was performed using an inbuilt function from the Sklearn library. The iteration stopped when the tolerance level reached 0.00001, with all other parameters set to default. This method estimates the missing values of a feature from all the other features. Non-integer values were rounded off where appropriate. For missing feature values, Mean Imputation was also performed, where missing values in a feature were replaced by the mean of the non-missing values in the same feature using an inbuilt function of Sklearn (Kılıç & Uğur, 2018). Non-integer values were rounded off as needed. The most frequent imputation technique was also attempted using the inbuilt function with default parameters for other missing data.

Modeling approach. LSTM is a widely used deep learning technique in RNN for time series prediction, not only for stock market prediction but also for classification and regression problems. The model employs a memory cell ct which is updated using input gate it, forget gate ft, and change gate ct, while the hidden state ht is updated using output gate ot and memory cell ct. The model takes a multivariate financial time series data from different sources as input and aims to predict the next day closing price using a multivariate sequence of input features (Chen, K., Zhou, Y., & Dai, F., 2015). To achieve this, the following procedures are followed: from the original dataset $X = (x1, x2,\dots, xn)$ of size $k{\times}n$, sequences $\{x1, x2,\dots, xn{-}1\}$ and $\{y1, y2,\dots, yn{-}1\}$ are created, where $xt \in$ R$k{\times}1$ is the input sequence and $yt \in$ R is the next day closing price at time t. Here, k and n represent the number of input features and the total number of observations, respectively. Input sequence Xt is created by taking m continuous sequences $xt : xt{+}m{-}1$, which is a matrix of shape $k \times m$ for $t \in \{1, 2,\dots, n - m - 1\}$, to incorporate the required dimension of LSTM architecture. The output ht of LSTM is a feature representation for the input sequence Xt at time t and is converted to a vector using a dense layer, since the final hidden state hf encodes the most information from the input sequence. Additionally, the average return Ravg is defined using equation (3), while the volatility $Vsec$ of a single asset is given by the formulas in equation (4). For portfolios with multiple assets, the volatility is calculated using the correlation coefficient and weights, as shown in equation (5) for a two-asset portfolio and the general formula in equation (6) for portfolios with more than two assets.

- $it = \sigma(Wixt + Whiht{-}1 + bi),$
  $ft = (Wf\,xt + Whf\,ht{-}1 + \ ),$
  $ot = (Woxt + Whoht{-}1 + bo),$

$ct = \tanh(Wcxt + Whcht{-}1 + bc),$
$ct = ft \otimes ct{-}1 + it \otimes ct,$
$ht = ot \otimes \tanh(ct)$ …………… (1)

The sigmoid and hyperbolic tangent functions are denoted by σ and tanh respectively. The element-wise product is represented by the operator $\otimes$. Weight matrices $W \in \mathrm{R}d \times k, h \in \mathrm{R}d \times d$, and bias vector $b \in \mathrm{R}d \times 1$ are also used. The sequence length, number of features, and hidden size are represented by n, k, and d respectively.

- $ht = (Xt, ht{-}1, ct{-}1,)$ ………… (2)

  Where w denotes all learnable parameters.

- $R_{avg} = \frac{\sum_{i=1}^{n} R_i}{n}$ ……….. (3)
- $V_{sec} = \sqrt{\frac{\sum_{i=1}^{n} Ri - R_{avg}}{n-1}}$ …………….. (4)
- $V_p = \sqrt{W_1{}^2\, v_1{}^2 + W_2{}^2\, v_2{}^2 + 2\, W_1 W_2\, v_1 v_2}$ ……………… (5)
- $V_P^2 = [W_1 \ldots\ldots\ldots W_n] \begin{bmatrix} V_{11} & \cdots & V_{1n} \\ \vdots & \ddots & \vdots \\ V_{n1} & \cdots & V_{nn} \end{bmatrix} \begin{bmatrix} W_{11} & \cdots & W_{1n} \\ \vdots & \ddots & \vdots \\ W_{n1} & \cdots & W_{nn} \end{bmatrix}$ ………………(6)

Model Creation

Splitting the dataset. To create the model, we partitioned the data into training and testing sets according to the time period/date when the stock prices were recorded. We developed a function that performed the partitioning by separating the dataset into training and testing subsets and subsequently dividing the training data into dependent and independent variables. We then further divided the data based on time, given that this is a time series problem.

Fitting the Model. We compiled the model using an optimizer imported from Keras, using MSE as the loss function and RMSE as the evaluation metric, which was defined earlier and also used from the Keras metric function. The model was fit using the training split of the dataset, with the dependent and independent variables passed as inputs. The validation dataset was used as the testing dataset, and a callback function was passed that we defined to implement the learning rate scheduler.

Performance Evaluation on Test Set. To assess the performance of the model, we first plot the curve for the actual stock prices and overlay it with the predicted values, as shown in Figure (9).

Fig 9

Curve for the actual stock prices with the predicted values

From this plot, we can see that the LSTM model is able to capture the trends in the stock prices to a certain extent. Additionally, it has accurately captured the recent dip in prices. In addition to visual inspection, we also calculated the RMSE and MAPE values to evaluate the model's performance, as previously decided. These values will be used for future comparisons. The LSTM model achieved even better results, as shown in Figure (10).

Fig 10

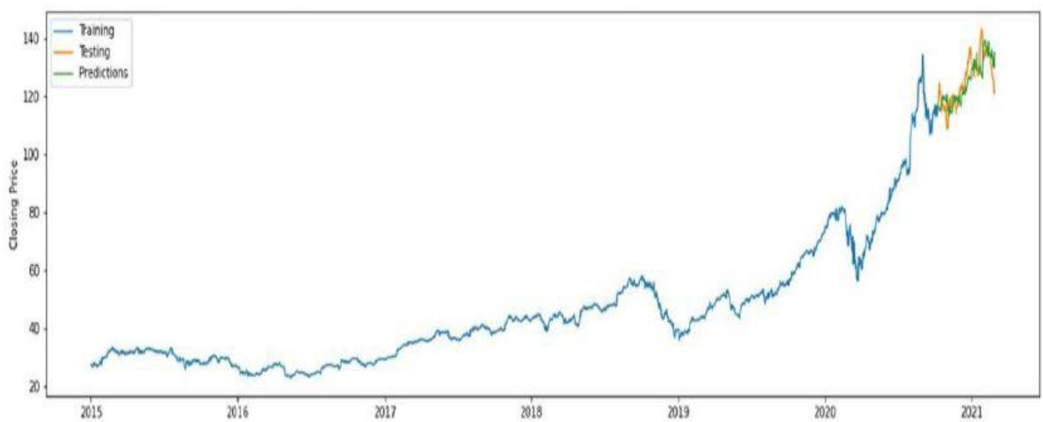
RMSE and MAPE Values

Results

The process of evaluating the performance of the trained model is crucial in determining its effectiveness. One way to visualize the results is through a graph that compares the predicted values with the actual values. In Figure (11), the train line, tested line, and predicted line are shown, and we can observe that the predicted values follow a similar trend as the actual values. However, it should be noted that predicting stock prices accurately is a complex task and there are various factors that can impact the fluctuations in stock prices, such as news and sudden changes in the market.

Fig 11

Training line, Tested line, and Predicted line



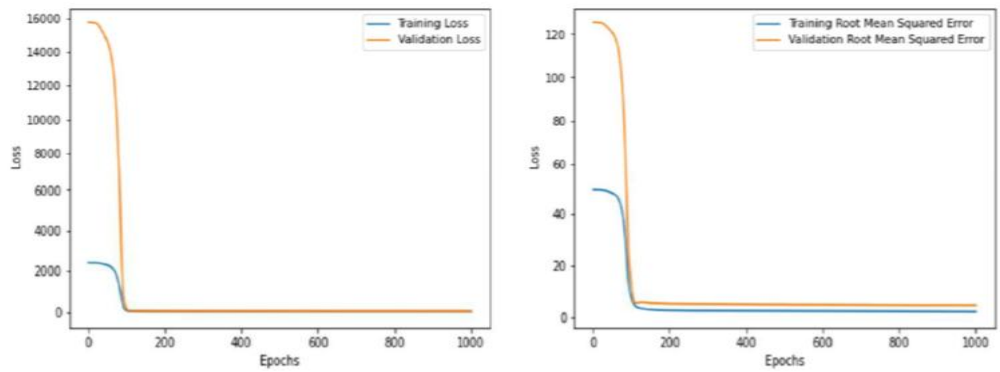
To further improve the model's performance, there are several approaches that can be considered, including increasing the training data, increasing the number of time steps, adding more LSTM layers or increasing the units in the LSTM layer, and incorporating additional indicators. These are just a few examples of the possible strategies for improving the model's accuracy.

Discussion and Analysis

Loss and Error.We kept a record of the loss and error, which decreased as the number of epochs increased. The figure (12) shows the loss and error for the model we created.

Fig 12

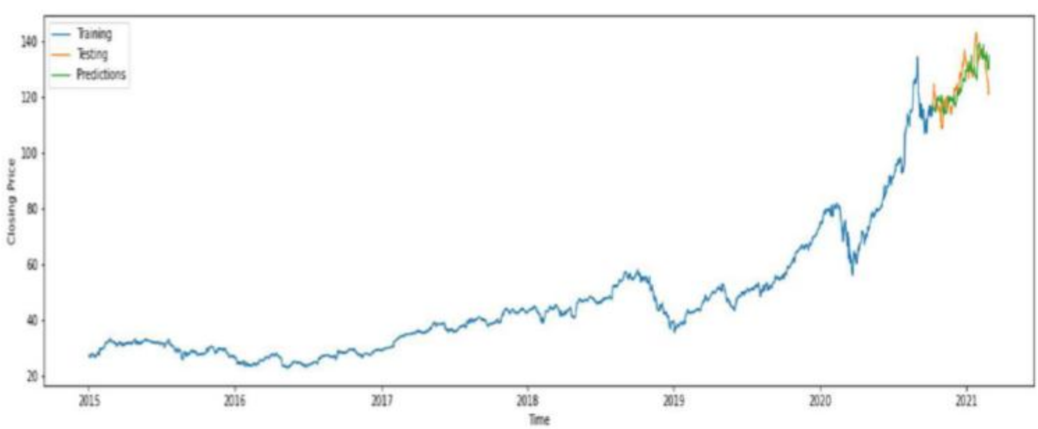
Loss and Error



Prediction of the Future Price Using Testing Dataset. To predict the future price, we utilized the predict function of the model which was compiled and fitted using the training dataset, and passed the testing dataset as input. The predicted and target values were plotted, and the results are shown in figure (13).

Fig 13

Training, Testing and predicted values



Conclusion

The popularity of stock market trading has been growing rapidly, which has led researchers to seek out new methods for prediction using novel techniques. Accurate forecasting techniques not only benefit researchers but also help investors or individuals dealing with the stock market. Our conclusion is that using machine learning to predict stock prices can increase prediction accuracy. By analyzing historical data, we can identify patterns and learn how stock prices have moved over the years. With a large dataset, machine learning models can learn these trends and predict future stock price movements. However, it's crucial to understand the parameters that influence the prediction to develop an efficient product with accurate results. While historical data is a powerful tool for forecasting, it's important to note that our method may not generate accurate results due to factors such as sudden changes in the market or positive/negative news.

Machine learning is an excellent way to identify patterns and make predictions more accurately than manual analysis of large data sets. However, one should not blindly rely on these models to make investment decisions. Deep learning models such as Long-Short-Term Memory (LSTM) Recurrent Neural Networks have proven to be advantageous over traditional machine learning methods in terms of accuracy and prediction speed. In this project, we applied the LSTM model for stock market prediction. The model's accuracy can be improved by training it on a greater number of data sets.

References

Althelaya, K. A., El-Alfy, E. S. M., & Mohammed, S. (2018, April). Evaluation of bi-directional LSTM for short-and long-term stock market prediction. *In 2018 9th International Conference on Information and Communication Systems* (pp. 151-156).

Bhalke, D. G., Bhingarde, D., Deshmukh, S., & Dhere, D. (2022). Stock price prediction using long short term memory. *A Journal of Physical Sciences, Engineering and Technology*, *14*(271-273).

Bhandari, H. N., Rimal, B., Pokhrel, N. R., Rimal, R., Dahal, K. R., & Khatri, R. K. (2022). Predicting stock market index using LSTM. *Machine Learning with Applications*, *9*, 100320.

Biswas, M., Shome, A., Islam, M. A., Nova, A. J., & Ahmed, S. (2021). Predicting stock market price: A logical strategy using deep learning. *In Symposium on Computer Applications & Industrial Electronics* (pp. 218-223).

Chen, K., Zhou, Y., & Dai, F. (2015, October). A LSTM-based method for stock returns prediction: A case study of china stock market. *International Conference on Big Data* (pp. 2823-2824).

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, *9*(8), 1735-1780.

Islam, S. B., Hasan, M. M., & Khan, M. M. (2021). Prediction of stock market using recurrent neural network. *Annual Information Technology, Electronics and Mobile Communication Conference* (pp. 479-483).

Kai, F., & Wenhua, X. (1997, October). Training neural network with genetic algorithms for forecasting the stock price index. *International Conference on Intelligent Processing Systems* (Vol. 1, pp. 401-403).

Kilic, D. K., & Ugur, O. (2018). Multi-resolution analysis of S&P500 time series. *Annals of Operations Research*, *260*, 197-216.

Kim, J., Hwang, M., Jeong, D. H., & Jung, H. (2012). Technology trends analysis and forecasting application based on decision tree and statistical feature analysis. *Expert Systems with Applications*, *39*(16), 12618-12625.

Moghar, A., & Hamiche, M. (2020). Stock market prediction using LSTM recurrent neural network. *Procedia Computer Science*, *170*, 1168-1173.

Naik, N., & Mohan, B. R. (2019). Optimal feature selection of technical indicator and stock prediction using machine learning technique. *In Emerging Technologies in Computer Engineering* (pp. 261-268).

Ortega, L., & Khashanah, K. (2014). A neuro-wavelet model for the short-term forecasting of high-frequency time series of stock returns. *Journal of Forecasting*, *33*(2), 134-146.

Pramod, B. S., & Pm, M. S. (2020). Stock price prediction using LSTM. *Test Engineering and Management*, *83*, 5246-5251.

Roondiwala, M., Patel, H., & Varma, S. (2017). Predicting stock prices using LSTM. *International Journal of Science and Research (IJSR)*, *6*(4), 1754-1756.

Shah, D., Isah, H., & Zulkernine, F. (2019). Stock market analysis: A review and taxonomy of prediction techniques. *International Journal of Financial Studies*, *7*(2), 26.

Staudemeyer, R. C., & Morris, E. R. (2019). Understanding LSTM-a tutorial into long short-term memory recurrent neural networks. *arXiv preprint arXiv:1909.09586*.

Sunny, M. A. I., Maswood, M. M. S., & Alharbi, A. G. (2020, October). Deep learning-based stock price prediction using LSTM and bi-directional LSTM model. *Novel Intelligent and Leading Emerging Sciences* (pp. 87-92).