Analytical detection of " Smart Stock Trading System" utilizing AI-model

Paramita Chowdhury[1*], Alaska[2], Subhradeep Nath[3], Sugata Das[4], Pradipta Patra[5]

[1*]Netaji Subhash Engineering College, Dept of Electronics and Communication Engineering, Techno City ,Garia,Kolkata-700152

*Corresponding Author: paramitamamon_aec@yahoo.com

***Abstract*— In the realm of finance, stock trading stands out as one of the most significant activities. Predicting stock prices involves analyzing available market data to forecast the future value of a particular stock or share. In our research, we developed a machine learning model along with LSTM (Long Short-Term Memory) to predict stock prices. Our model follows the 70:30 principle, allowing the model to improve its accuracy over time. We collected a decade's worth of data from Yahoo Finance and employed customized feature engineering and deep learning techniques to analyze the stock market trends. Our approach has shown promising accuracy in predicting stock market trends. By carefully optimizing prediction term lengths, feature engineering methods, and data preprocessing techniques, our work contributes to advancing research in both financial analysis and technical modeling within the stock market domain.**

Keywords— LSTM, Machine learning, RNN, Data Prediction, finance, linear regression.

INTRODUCTION

For decades, researchers have delved into the complexities of stock market predictions. However, due to its inherent intricacies, constant flux, and chaotic nature, this task has proven immensely challenging. The multitude of variables and information sources involved, coupled with a significant signal-to-noise ratio, exacerbates the difficulty of accurately forecasting market behavior. Despite ongoing scientific discourse on the feasibility of such predictions, the majority of models fall short of providing precise insights. Nevertheless, numerous studies across diverse disciplines endeavor to tackle this challenge, offering a myriad of approaches in pursuit of this goal. One prevalent strategy involves leveraging Machine Learning algorithms to analyze historical price data for future price projections. This article explores a specific method employing recurrent neural networks, particularly focusing on their short-term memory capacity. The chosen algorithm, the LSTM (Long Short-Term Memory) network, distinguishes itself by effectively discerning between recent and distant examples, assigning varying weights to each while discarding irrelevant memory to predict subsequent outputs. This characteristic enables LSTM networks to handle longer input sequences compared to their counterparts limited to shorter sequences. Utilizing historical candlestick data from various stocks on the Brazilian stock exchange (Bovespa), along with an array of technical indicators as features, this study trains and evaluates the LSTM model. The model aims to predict whether the price of a specific stock will rise within the next 15 minutes. The overarching objective is to assess the efficacy of recurrent neural networks, specifically LSTM networks, in forecasting stock market movements, gauging their performance against traditional machine learning algorithms using real-life data. Furthermore, the model's financial performance is validated by comparing it with straightforward yet effective investment strategies in terms of overall returns and per-trade net results. The key contributions of this research include the development of a novel stock market price movement prediction model utilizing deep learning techniques, its validation using real data from the Brazilian stock exchange, and its evaluation against established baselines. The CNX NIFTY, NSE's flagship index, is widely utilized by investors both domestically and globally. The exchange facilitates trading, settlement, and clearing in equity, debt, and derivatives markets, making it one of the largest trading platforms worldwide. Various domestic and international companies, including TATA, WIPRO, HDFC, and YES BANK Ltd., have vested interests in the exchange.

Kim et al. built a model as a combination of artificial neural networks (ANN) and genetic algorithms (GAs) with discretization of features for predicting stock price index. The data used in their study include the technical indicators as well as the direction of change in the daily Korea stock price index (KOSPI). They used the data containing samples of 2928 trading days, ranging from January 1989 to December 1998, and give their selected features and formulas. They also applied optimization of feature discretization, as a technique that is similar to dimensionality reduction. The strengths of their work are that they introduced GA to optimize the ANN. First, the amount of input features and processing elements in the hidden layer are 12 and not adjustable. Another limitation is in the learning process of ANN, and the authors only focused on two factors in optimization. While they still believed that GA has great potential for feature discretization optimization. [1]. Nusrat et al. conducted research analyzing machine learning algorithms, particularly linear regression, for stock price prediction. They emphasized the significance of feature selection and data pre-processing in enhancing prediction model accuracy [2]. Payal et al. examined state-of-the-art machine learning techniques for stock price prediction, including linear regression. They underscored the importance of integrating external factors like news sentiment and macroeconomic indicators into prediction models. [3]. Hassan et al. in applied the Hidden Markov Model (HMM) on the stock market forecasting on stock prices of four different Airlines. Tey reduce states of the model into four states: the opening price, closing price, the highest price, and the lowest price. The strong point of this paper is that the approach does not need expert knowledge to build a prediction model. While this work is limited within the industry of Airlines and evaluated on a very small dataset, it may not lead to a prediction model with generality. One of the approaches in stock market prediction related works could be exploited to do the comparison work. The authors selected a maximum of 2 years as the date range of training and testing dataset, which provided us a date range reference for our evaluation part. [4]. Lei et al.in exploited Wavelet Neural Network (WNN) to predict stock price trends. The author also applied Rough Set (RS) for attribute reduction as an optimization. Rough Set was utilized to reduce the stock price trend feature dimensions. It was also used to determine the structure of the Wavelet Neural Network. The dataset of this work consists of five well-known stock market indices, i.e., (1) SSE Composite Index (China), (2) CSI 300 Index (China), (3) All Ordinaries Index (Australian), (4) Nikkei 225 Index (Japan), and (5) Dow Jones Index (USA). The evaluation of the model was based on different stock market indices, and the result was convincing with generality. Using Rough Set for optimizing the feature dimension before processing reduces the computational complexity. However, the author only stressed the parameter adjustment in the discussion part but did not specify the weakness of the model itself. Meanwhile, we also found that the evaluations were performed on indices, the same model may not have the same performance if applied on a specific stock. [5]. Lee et al. in used the support vector machine (SVM) along with a hybrid feature selection method to carry out prediction of stock trends. The dataset in this research is a sub dataset of NASDAQ Index in Taiwan Economic Journal Database (TEJD) in 2008. The feature selection part was using a hybrid method, supported sequential forward search (SSFS) played the role of the wrapper. Another advantage of this work is that they designed a detailed procedure of parameter adjustment with performance under different parameter values. The clear structure of the feature selection model is also heuristic to the primary stage of model structuring. One of the limitations was that the performance of SVM was compared to back-propagation neural network (BPNN) only and did not compare to the other machine learning algorithms.[6] Indu et al. assessed the effectiveness of different machine learning techniques, including Forestry and Support Vector Machines, for stock price prediction. [7]. Kara et al.in also exploited ANN and SVM in predicting the movement of stock price index. The data set they used covers a time period from January 2, 1997, to December 31, 2007, of the Istanbul Stock Exchange. The primary strength of this work is its detailed record of parameter adjustment procedures. While the weaknesses of this work are that neither the technical indicator nor the model structure has novelty, and the authors did not explain how their model performed better than other models in previous works. Thus, more validation works on other datasets would help. They explained how ANN and SVM work with stock market features, and also recorded the parameter adjustment. The implementation part of our research could benefit from this previous work. [8]. Long et al. conducted a deep learning approach to predict the stock price movement. The dataset they used is the Chinese stock market index CSI 300. For predicting the stock price movement, they constructed a multi-filter neural network (MFNN) with stochastic gradient descent (SGD) and back propagation optimizer for learning NN parameters. The strength of this paper is that the authors exploited a novel model with a hybrid model constructed by different kinds of neural networks, it provides an inspiration for constructing hybrid neural network structures. [9]. Thakur et al.in also developed a hybrid financial trading support system by exploiting multi-category classifiers and random forest (RAF). They conducted their research on stock indices from NASDAQ, DOW JONES, S&P 500, NIFTY 50, and NIFTY BANK. The authors proposed a hybrid model combined random forest (RF) algorithms with a weighted multicategory generalized eigenvalue support vector machine (WMGEPSVM) to generate “Buy/Hold/Sell” signals. Before processing the data, they used Random Forest (RF) for feature pruning. The authors proposed a practical model designed for real-life investment activities, which could generate three basic signals for investors to refer to. They also performed a thorough comparison of related algorithms. While they did not mention the time and computational complexity of their works. Meanwhile, the unignorable issue of their work was the lack of financial domain knowledge background. The investors regard the indices data as one of the attributes but could not take the signal from indices to operate a specific stock straight forward. [10].

The novelty of our research work lies in leveraging the Raspberry Pi 4 to create a compact and portable solution for stock prediction plus stock trading and monitoring. By utilizing the Raspberry Pi's capabilities, you're able to integrate three display ports, enabling the use of multiple displays with a single device. This multi-display functionality is innovative, as it streamlines the trading process and enhances efficiency for users. Additionally, your project offers a cost-effective solution by eliminating the need for heavy investment in multiple devices, making it accessible to a wider audience. Users can conveniently monitor, buy, and check stock predictions all from one centralized platform, enhancing convenience and accessibility in the world of stock trading. Combining trading functionality with stock prediction on a compact and portable device like the Raspberry Pi 4 offers a comprehensive solution for both monitoring the market and executing trades. This integration of machine learning and hardware technology represents a cutting- edge approach to personal finance and investment management, setting your project apart as a forward- thinking and innovative solution in the realm of stock trading and prediction.

International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 08 Issue: 05 | May - 2024 SJIF Rating: 8.448 ISSN: 2582-3930

EXPERIMENTAL METHODOLOGY AND SETUP

A. Recurrent Neural Network (RNN):

Recurrent Neural Network (RNN) is a specific type of neural network architecture where the output from the previous step is utilized as input for the current step. Unlike traditional neural networks where inputs and outputs are treated independently, RNNs are adept at predicting subsequent words in a sentence by retaining information from previous words. This capability is facilitated by a Hidden Layer within RNNs. The pivotal aspect of RNNs lies in their Hidden state, which preserves sequence-related information, often referred to as Memory State. By employing the same parameters across inputs or hidden layers, RNNs effectively streamline complexity, distinguishing them from other neural network architectures.

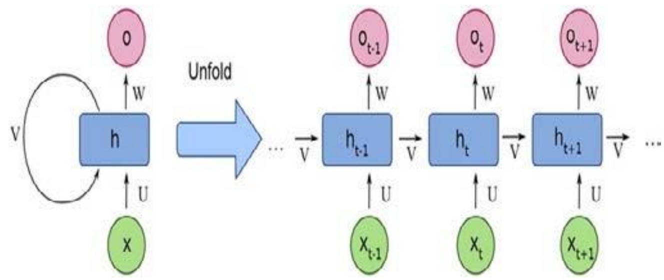


Fig 1: Recurrent Neural Network

B. PROPOSED TECHNOLOGY:

Deep learning models are highly favored across various domains of science and engineering. They have gained significant traction in stock price forecasting and trend prediction, primarily because of their capacity to decipher intricate patterns, manage extensive datasets, facilitate feature learning and representation, and adjust to dynamic market conditions.

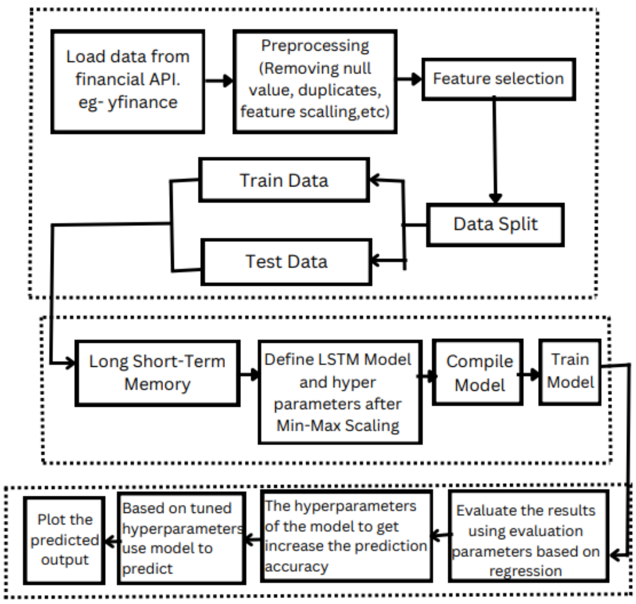


Fig 2: Workflow of basic ML Model

Training the ML model comprises six steps, as outlined in the following section:

Step 1: Begin by loading the data from a CSV file or retrieving historical data through an API, such as Yahoo Finance. Step 2: Preprocess the historical data to eliminate redundancies, handle null values, and conduct feature selection. Step 3: Prior to training the ML model, select relevant features like open, close, adjusted close, volume, etc., alongside any secondary data. Step 4: Split the preprocessed data into training and testing sets, typically allocating 70% for training and 30% for testing to evaluate model performance. Step 5: Utilize the training data to train the model, followed by evaluating its performance using the testing data. Step 6: Once the model is trained, assess its performance using evaluation parameters for regression or classification tasks. Step 7: Fine-tune the model's hyperparameters to enhance evaluation parameters. Evaluate the model post-tuning to verify any improvements in prediction accuracy and visualize the predictions accordingly.

C. Long Short Term Memory (LSTM) Network:

Long Short-Term Memory (LSTM) represents an advancement in recurrent neural network technology pioneered by Hochreiter & Schmid Huber. Renowned for its efficacy in sequence prediction tasks, LSTM excels in capturing prolonged dependencies within data sequences. Its utility extends across various domains, including time series analysis and sequence-based applications like machine translation and speech recognition. This overview delves into LSTM's intricacies, detailing its model, architecture, operational principles, and its pivotal role across diverse applications. Unlike conventional RNNs, which struggle with learning long-term dependencies due to a single hidden state passed through time, LSTMs tackle this challenge by introducing a memory cell. This specialized component retains information over extended durations, facilitating the learning of intricate patterns within sequential data. As a result, LSTMs prove invaluable in tasks such as language translation, speech recognition, and time series forecasting. Moreover, LSTMs seamlessly integrate with other neural network architectures, such as Convolutional Neural Networks (CNNs), enhancing capabilities in tasks like image and video analysis. Central to LSTM's functionality are three gates: the input gate, the forget gate, and the output gate. These gates regulate the flow of information within the memory cell, determining what information to store, discard, or output. By selectively managing data flow, LSTM networks adeptly preserve pertinent information while discarding irrelevant details, enabling effective learning of long-term dependencies.

International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 08 Issue: 05 | May - 2024 SJIF Rating: 8.448 ISSN: 2582-3930

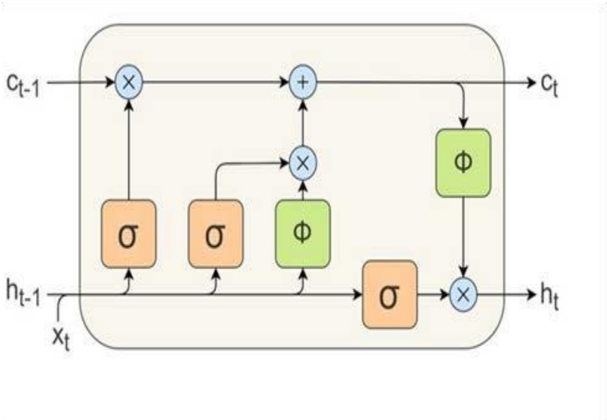


Fig 3: Illustrates the internal architecture of an LSTM network.

D. LSTM Architecture:

As shown in Fig. 3, the input to the current cell state (Ct) is the previous hidden state (ht-1), previous cell state (Ct-1) and present input (Xt). The cell consists of three gates i.e. forget gate, input gate and output gate.

Forget Gate: A forget gate will remove unnecessary data from the cell state. The information that is less important or not required for the LSTM to understand things is removed by performing multiplication of hidden state by a sigmoid function. This step is necessary to optimize the performance of the model. It takes two inputs i.e., h(t-1) and xt , where h(t-1) is the previous cell hidden state output and xt is the current cell input.

$$Ft = \sigma\,(Wfx * Xt + Wfh * ht\text{-}1 + bf) \quad (1)$$

Input Gate: This cell is responsible for regulating the data that is added to the cell from the input. Forget gate is used to filter some input. A vector is created by adding all the possible values from the previous cell hidden state h(t-1) and current cell input Xt by using the tanh function. The output of the tanh function in the ranges of [-1, 1]. Finally, the outputs of sigmoid and tanh functions are multiplied, and the output is added to the cell state.

$$It = \sigma\,(Wix * Xt + Whh * ht\text{-}1 + bi) + \tanh(Wcx * Xt + Wch * ht-1 + bi) \quad (2)$$

Output Gate: Tanh function is applied to the cell state to create a vector with all possible values. Sigmoid function is applied to previous cell hidden state h(t-1) and current cell input xt to filter necessary data from the previous cell. Now, the outputs of sigmoid and tanh functions are multiplied and this output is sent as a hidden state of the next cell.

$$Ot = \sigma\,(Wox * Xt + Whh * ht\text{-}1 + Woc * Ct\text{-}1 + bi) \quad (3)$$

Intermediate cell state (Ct) is obtained by the multiplication of Forget gate (Ft) with previous cell state (Ct-1). Then this intermediate state is added to the output of the input gate.

$$Ct = Ft * Ct\text{-}1 + It \quad (4)$$

The current hidden/output state is obtained by multiplying output gate and tanh of cell state.

$$ht = Ot * \tanh(Ct) \quad (5)$$

E. Neural Prophet (NP):

Prophet is a time series forecasting model developed by Facebook's Core Data Science Team. It comprises various components including trend, seasonality, auto-regression, and additional regressors. The main model components of Prophet consist of trend, seasonality, and holidays, which are integrated using equation.

$$y(t)=g(t)+s(t)+h(t)+e(t)y(t)=g(t)+s(t)+h(t)+e(t) \quad (6)$$

Here, $g(t)g(t)$ is a trend-modeling function that can be specified as a linear function or a logistic function. $s(t)s(t)$ represents a seasonality function that can be daily, weekly, and/or yearly, which is handled with Fourier terms. $h(t)h(t)$ is a holiday function that considers the effect of holidays, which occur irregularly. $e(t)e(t)$ represents the error changes that are not fitted by the model.

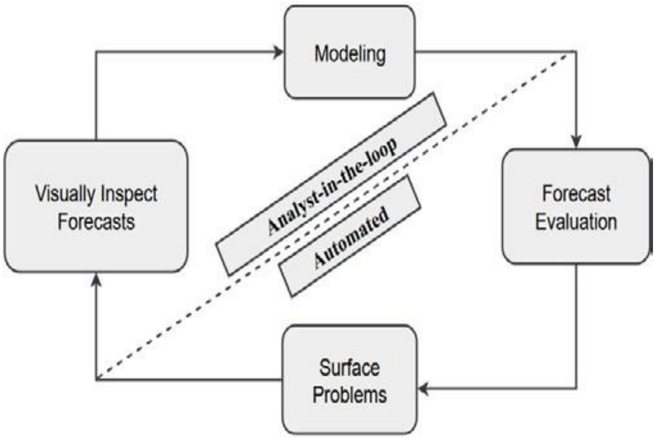


Fig 4: Forecasting process in FbProphet & neural prophet.

F. HYBRID MODEL ARCHITECTURE:

Yahoo Finance is used as the input source for the data. The information is made up of stock information for a number of product- and service-based businesses, which displays the market closing rates for each of their individual stocks. For preprocessing, the data is divided into two unique sections: training data and testing data. The machine learning model's input data will be the bifurcated data. The data is utilized to build an LSTM model, which will be used to forecast stock value. The model's correctness is then verified by a number of rounds to make sure the strategy used resolves the issues that were in the prior models. Before the needed accuracy is attained in reliable numbers, the procedure proceeds through multiple iterations. Once the model has received enough training, streamlit is used to build a web application that gives the user the chance to use the model and forecast the data of their invested stocks. The user can choose between companies that offer products and services using the radio buttons provided by the web app. Additionally, a forecast for the next 30 days at most can be obtained. A slider is offered for this purpose, allowing the prediction to be changed from one day to a whole month. With the help of this specific feature, the investor would be able to decide whether to keep the stock in the market, purchase additional shares of related companies, or sell it in order to cut losses. Utilizing Google Collab and Streamlit, the Stock Market Prediction Using LSTM (Web Application) is effectively implemented.

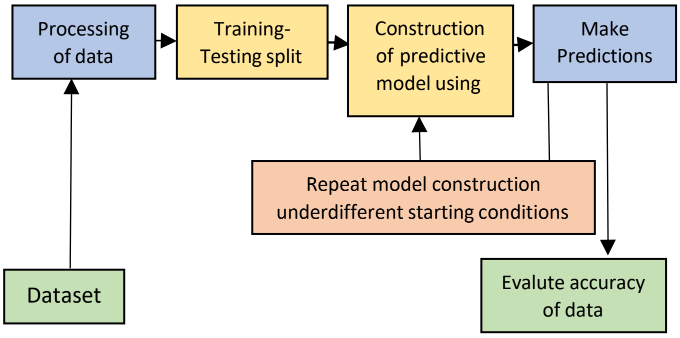


Fig 5: Workflow of hybrid model architecture

Data Selection: The first step is to select data for an organization and split the data into training and testing. We have used 70% for training and 30% for testing purposes.

Pre-processing of data: In pre-processing, we are selecting attributes required for the algorithm and the remaining attributes are neglected. The selected attributes are Trade Open, Trade High, Trade Low, Trade Close, Trade Volume. In pre-processing, we are using normalization to get values in a particular range.

Prediction using LSTM: In this system, we are using the LSTM algorithm for predicting stock values. Initially, the training data is passed through the system and trains the model. Then in the testing phase, the predicted values are compared with the actual values.

Evaluation: In the evaluation phase we are calculating the Accuracy, Mean Square Error (MSE) and Root Mean Square Error (RMSE) values for comparison.

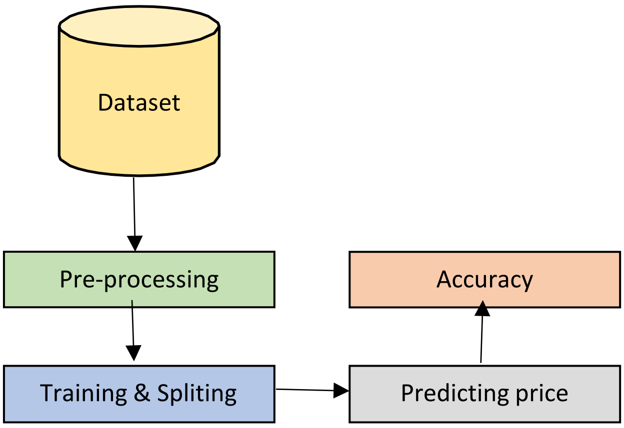


Fig 6: Diagram of dataset workflow

- **PARAMETERS USED**

List of parameters/Symbols used in this paper is listed in Table 1

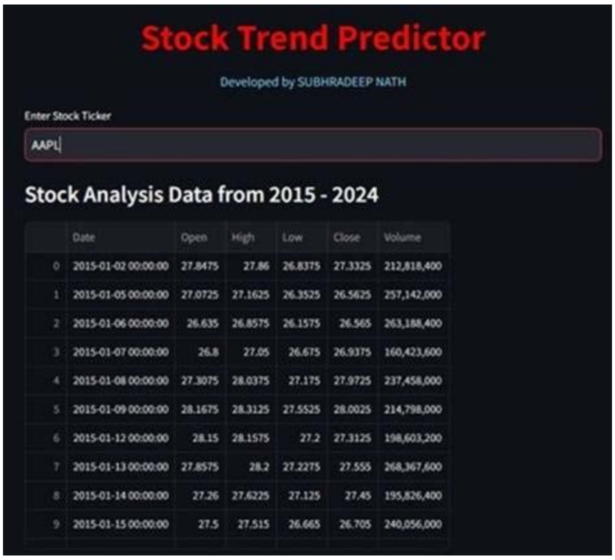
Table 1: Parameters Used

Parameter Used	Meaning
Date	Date of stock price
Open	Open price of a share
Close	Closing price of a share
Volume/ trade	Quantity Number of shares traded
High	Highest share value for the day
Low	Lowest share value for the day
Turnover	Total Turnover of the share

RESULT AND DATA ANALYSIS

The implementation of proposed LSTM model using python which predicts the future price of AAPL share based on its historical data. The below visualization figure shows the visualization of AAPLSHARE prediction. In our paper, the implementation of an algorithm which predicts the stock price of a share for a given period of time, the graph below from our algorithm will show the predicted price of AAPL shares. In the result shown in the below graph is the plotted form our algorithm outcome by applying 96 LSTM units for achieving the accuracy.

The ticker symbol for the stock to be analyzed is defined (here, 'AAPL' for Apple Inc.). Historical stock data for the specified ticker symbol is downloaded from Yahoo Finance, covering the time period from January 1, 2015, to April 10, 2024.



	Date	Open	High	Low	Close	Volume
0	2015-01-02 00:00:00	27.8475	27.86	26.8375	27.3325	212,818,400
1	2015-01-05 00:00:00	27.0725	27.1625	26.3525	26.5625	257,142,000
2	2015-01-06 00:00:00	26.635	26.8575	26.1575	26.565	263,188,400
3	2015-01-07 00:00:00	26.8	27.05	26.675	26.9375	160,423,600
4	2015-01-08 00:00:00	27.3075	28.0375	27.175	27.9725	237,458,000
5	2015-01-09 00:00:00	28.1675	28.3125	27.5525	28.0025	214,798,000
6	2015-01-12 00:00:00	28.15	28.1575	27.2	27.3125	198,603,200
7	2015-01-13 00:00:00	27.8575	28.2	27.2275	27.555	268,367,600
8	2015-01-14 00:00:00	27.26	27.6225	27.125	27.45	195,826,400
9	2015-01-15 00:00:00	27.5	27.515	26.665	26.705	240,056,000

Fig 7: Stock Analysis Data

The moving average is a commonly used technical analysis tool to smooth out price data by creating a constantly updated average price. The 100-day moving average (MA100) and

International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 08 Issue: 05 | May - 2024 SJIF Rating: 8.448 ISSN: 2582-3930

200-day moving average (MA200) are calculated for the closing prices of the stock data. These moving averages provide insights into the longer-term trends in the stock price by averaging out short-term fluctuations.

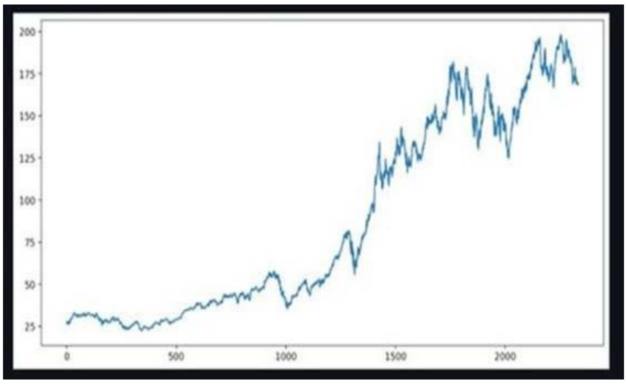


Fig 8: Closing price & Time chart

100-day Moving Average (MA100): The 100-day moving average is calculated by taking the average of the closing prices over the previous 100 days. This moving average helps smooth out short-term fluctuations in the stock price and highlights the underlying trend over a medium-term period. A rising MA100 indicates that the stock price is generally increasing over the past 100 days, while a falling MA100 suggests a downward trend.

Fig 9: 100 days moving average

200-day Moving Average (MA200): Similarly, the 200-day moving average is calculated by averaging the closing prices over the previous 200 days. This moving average provides a longer-term perspective on the stock price trend and is often used by investors to identify major shifts in the market sentiment. A crossover between the MA100 and MA200, where the MA100 crosses above the MA200, is often considered a bullish signal indicating a potential uptrend, while a crossover in the opposite direction may signal a downtrend.

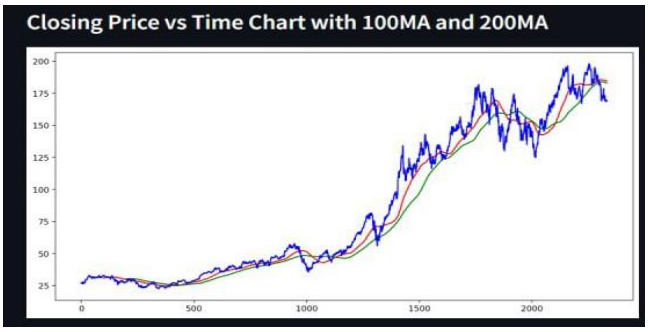


Fig 10: 200 days moving average

Splits the data into training and testing sets, where 70% of the data is for training and the remaining 30% is for testing. Imports MinMaxScaler from sklearn.preprocessing to normalize the data between 0 and 1. MinMaxScaler is a preprocessing technique used in machine learning to scale and normalize features within a specific range. It transforms the features by scaling them to a given range, typically between 0 and 1. MinMaxScaler is used in this code to normalize the stock price data before feeding it into the LSTM model. Normalization ensures that all input features contribute equally to the training process and prevents features with larger scales from dominating the learning process. Keras is used to define the LSTM model architecture and compile the model, while TensorFlow handles the execution of the computations defined in the model. Together, they provide a powerful framework for building, training, and deploying deep learning models, such as the LSTM model used for stock price prediction. Defines the LSTM model architecture using the Sequential API from Keras. The architecture includes four LSTM layers with dropout regularization. Compiles the model using the Adam optimizer and mean squared error loss function, then trains the model on the training data for 50 epochs.

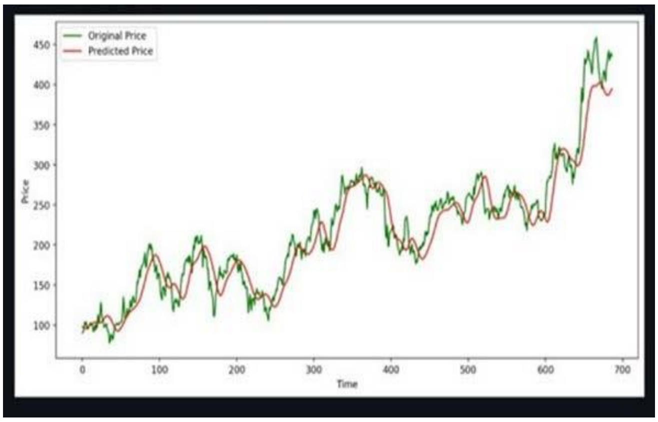


Fig 11: Original Price & Prediction Price

The file keras_model.h5 is a serialized representation of the trained LSTM model saved in the Hierarchical Data Format version 5 (HDF5) format. Selects the last 100 days of the training data and concatenates them with the testing data to create a new DataFrame for prediction. Normalizes the input data (concatenated DataFrame) using the Min-Max scaler. Prepares the testing data by creating sequences of 100 days with their corresponding target values for prediction. Makes predictions on the testing data using the trained LSTM model.

Scales back the predicted and actual values to their original scales using the inverse of the Min-Max scaling factor. Plots the actual and predicted stock prices over time for visualization.

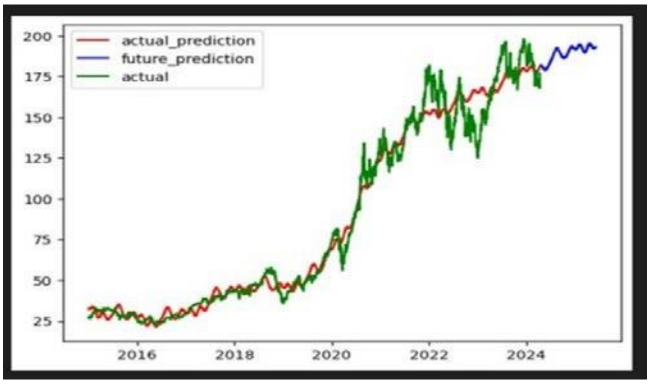


Fig 12: Future Prediction

CONCLUSION

This review examines various conventional, machine learning, and deep learning techniques utilized in stock market forecasting. It covers a range of methods including machine learning, deep learning, and time series forecasting. Recent applications of deep learning models are also tested using AAPL stock datasets. Despite the availability of numerous methods for stock price forecasting, there is still no universally effective solution for accurately predicting stock prices or market trends. It is crucial to efficiently train AI-based models with current data to avoid potential failures. In conclusion, researchers should continue exploring new avenues, particularly leveraging ensemble techniques, to address price action challenges in stock forecasting. Enhancements in stock forecasting models through suitable hyperparameter tuning are essential to ensure precise predictions. While machine learning and deep learning models can serve as valuable supplementary indicators for traders and investment advisors, decisions should not solely rely on AI-based forecasting methods. Furthermore, future research endeavors could expand into areas such as portfolio management, trading strategy formulation, and investment decision-making.

ACKNOWLEDGMENT

We thank the Electronics and Communications Engineering Department of Netaji Subhash Engineering College for providing us with the necessary resources which were absolutely essential for undertaking such a project to design the model.

REFERENCES

[1] Kim K, Han I. Genetic algorithms approach to feature discretization in artificial neural networks for the prediction of stock price index. Expert Syst Appl. 2000;19:125–32. https://doi.org/10.1016/S0957-4174(00)00027-0.

[2] Rouf N. and Singh’s S. “Stock Price Prediction Using Machine Learning Techniques: A Decade Survey on Methodologies, Recent Developments, and Future Directions” MDPI. Published in November 2021. https://doi.org/10.3390/electronics10212717

[3] Soni P., Tewari Y., Krishnan D. “Machine Learning approaches in Stock Price Prediction: A Systematic Review”, Journal of Physics Conference Series. Published in 2022. https://doi.org/10.1088/1742-6596/2161/1/012065

[4] Hassan MR, Nath B. Stock market forecasting using Hidden Markov Model: a new approach. In: Proceedings—5th international conference on intelligent systems design and applications 2005, ISDA’05. 2005. pp. 192–6. https://doi.org/10.1109/ISDA.2005.85

[5] Lei L. Wavelet neural network prediction method of stock price trend based on rough set attribute reduction. Appl Soft Compute J.2018;62:923–32. https://doi.org/10.1016/j.asoc.2017.09.029.

[6] Lee MC. Using support vector machine with a hybrid feature selection method to the stock trend prediction. Expert Syst Appl. 2009;36(8):10896–904. https://doi.org/10.1016/j.eswa.2009.02.038.

[7] Kumar I., Utreja’s C. “A Comparative Study of Supervised Machine Learning Algorithms for Stock Market Trend Prediction”. IEEE Conference. (2018). https://doi.org/10.1109/ICICCT.2018.8473214

[8] Kara Y, Acar Boyacioglu M, Baykan ÖK. Predicting direction of stock price index movement using artificial neural networks and support vector machines: the sample of the Istanbul Stock Exchange. Expert Syst Appl. 2011;38(5):5311–9. https://doi.org/10.1016/j.eswa.2010.10.027.

[9] Long W, Lu Z, Cui L. Deep learning-based feature engineering for stock price movement prediction. Knowledge Based Syst. 2018;164:163–73. https://doi.org/10.1016/j.knosys.2018.10.034.

[10] Thakur M, Kumar D. A hybrid financial trading support system using multi-category classifiers and random forest. Appl Soft Compute J. 2018; 67:337–49. https://doi.org/10.1016/j.asoc.2018.03.006.