Stock Prediction Using Deep Learning

Models LSTM, Bi-LSTM,GRU and

XGBOOST

Aashmit Shrestha

*Computer Science and Engineering,*

*Kalinga Institute of Industrial Technology*

Bhubaneswar,India

aashmit13@gmail.com

*Abstract*—Owing to its volatility and its dependence on numerous factors predicting the future course of the stock market is difficult. Forecasts are frequently made using technical indicators and historical data to address this issue. A few examples of Deep Learning RNNs that may be utilized to take advantage of various properties to improve the prediction accuracy include LSTM, Bidirectional LSTM, GRU, and XGBOOST. This study aims to improve the strategy and increase the accuracy of forecasts.Historical data as well as technical indicators such as the Moving Average make the input data. The framework proposed in this study uses GRU to identify long-term trends and XGBoost to generate final forecasts.

Keywords—Bidirectional LSTM, GRU, LSTM, XGBoost, Technical Indicators, Stock Prediction,RNN

# **Introduction**

The stock market is characterized by its erratic and dynamic nature, where prices may change promptly due to a range of factors, including geopolitical events, economic indicators, and corporate performance. Accurately forecasting stock prices poses a challenge for investors, traders, and financial analysts. Machine learning algorithms have become increasingly popular to predict stock prices because of their capability to analyze vast amounts of data and detect intricate patterns in the given datasets.

The analysis of sequential data using a neural network architecture involves the usage of Recurrent Neural Networks (RNN). The classification of RNN that has gained popularity in stock market prediction owing to its ability to detect long-term dependencies in sequential data is LSTM. In stock market prediction, LSTM models are trained on historical stock prices and related features to learn the patterns in the data. These patterns were then used to predict future stock prices. The LSTM model has a memory unit that can remember important information for a longer period of time, allowing it to capture long-term trends and patterns in stock market data. LSTM models have shown promising results in previous studies and are widely used in the financial industry for stock market prediction.

The dataset used in this study is the historical stock price data of a selected company, including features like closing price,high price,opening price, low price and volumes. The dataset was preprocessed by normalizing the values and splitting them into testing and training sets.

The training set was used to train the LSTM, Bi-LSTM, and GRU models, while the testing set was used to evaluate them. These models are categories of Recurrent Neural Networks (RNN). They posses the ability to capture long-term dependencies in sequential data. The The Bi-LSTM model is capable of handling data in both forward and backward directions, whereas the LSTM model has memory units that can retain significant information for a longer period of time. The GRU model has lesser parameters in comparison to the LSTM model, making it faster to train and execute.

The XGBoost model is categorized as a Gradient Boosting Machine (GBM) and operates by utilizing decision trees to generate predictions. This approach involves the gradual introduction of new trees to correct errors made by previous trees. The XGBoost model is trained on a similar set of features as RNN models using the training set, and subsequently assessed using the testing set.The proposal presented in this is for a hybrid deep learning model that utilizes historical data to forecast stock prices.Contributions of proposed works are:

* Selection of befitting technical indicator
* Integration of GRU with XGBoost technique for final stock price prediction.

# **PROPOSED WORK**

In this study, we compare the performance and accuracy of popular and frequently used deep learning algorithms for stock market prediction: Bidirectional LSTM (Bi-LSTM),Long Short-Term Memory (LSTM), Gated Recurrent Unit (GRU), combined bidirectional LSTM with extreme gradient boosting (GRU-XGBoost), and propose a hybrid model combining GRU with Extreme Gradient Boosting (GRU-XGBoost).

The dataset used in this study is the historical stock price data of Apple, including the features such as the closing price, high price,opening price,low price, and volume. The values in the dataset were normalised and split into training and testing sets as part of the preprocessing.

The training set was used to train the Bi-LSTM, GRU, LSTM, Bi-LSTM XGBoost and GRU-XGBoost models, which were then assessed on the testing set. These models are types of Recurrent Neural Networks (RNN) Thanks to its memory cell, the LSTM (Long Short-Term Memory) model is a kind of recurrent neural network that is capable of maintaining important data for a longer amount of time. With the use of this memory cell, the LSTM is able to avoid the vanishing gradient issue while retaining essential information across a long data set. To put it another way, the LSTM can spot recurring dependencies in the input sequence.In contrast, the Bi-LSTM model has the capacity to process data in both forward and backward directions, enabling it to recognize relationships in both the past and future situations.Meanwhile, the GRU (Gated Recurrent Unit) model is another sort of RNN that accomplishes comparable tasks to the LSTM but with less parameters, allowing it to train quicker. It is more computationally effective than the LSTM model because it makes use of reset and update gates to regulate the information flow.The RNN and Gradient Boosting Machine (GBM) models' best qualities are combined in the hybrid GRU-XGBoost model.

The effectiveness of each model was assessed using measures for evaluating accuracy. The Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), Coefficient of Determination (R2) and Mean Absolute Percentage Error (MAPE) was calculated for all of the models. The average squared difference between the expected and actual values is what the RMSE calculates. The average absolute difference between the expected and actual values is measured by MAE. The average percentage difference between the expected and actual values is calculated using the MAPE. The proportion of accurate predictions the model makes is gauged using the R2 statistic.

The experimental results were analyzed to determine the model that performed best in terms of accuracy and error metrics. The findings of this study provide insights into which machine learning algorithm is most suitable for predicting stock prices and can be useful for investors, traders, and financial analysts.

Open, High, Low, Closing,Volume were derived from the dataset. The GRU-XGBoost hybrid model combines the strengths of both RNN and Gradient Boosting Machine (GBM) models. The GRU model extracts features from the data, and the features are fed into the XGBoost model for the final prediction. The GRU-XGBoost hybrid model's performance was assessed using the same metrics as the Bi-LSTM, LSTM and GRU models. The framework for the proposed model is as follows:

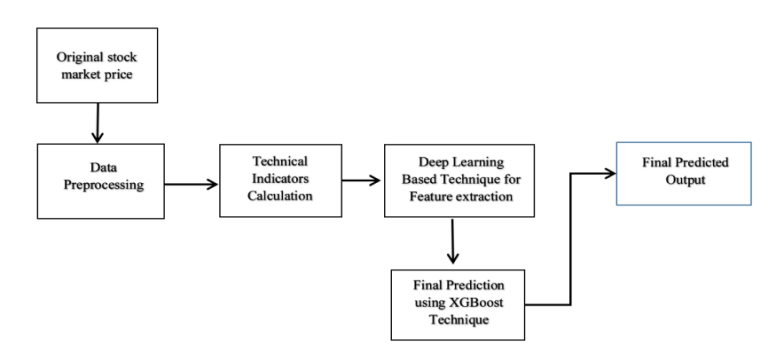
.

Fig1: Flowchart of the proposed GRU-XGBoost Model

The dataset was extracted from Yahoo Finance and the 10 year time period between 2013-01-01 and 2023-01-01 was utilized for this work..

## Long-Short-Term Memory (LSTM)

For time series prediction applications, the LSTM (Long Short-Term Memory) Recurrent Neural Network architecture is frequently used. Being able to choose recall or ignore previous information, this style of architecture is very successful when processing time-series data with long-term dependencies. A series of past prices and other pertinent variables, such as trade volume, market indices, and sentiment towards the news, are commonly represented as the input data in a stock price prediction model using LSTM. Following this series of inputs processing, the LSTM model produces an output that reflects the forecasted stock price at a specific timestep.

The LSTM architecture consists of three main components:

1. Forget gate: In an LSTM neural network cell, the first step is to determine whether we should retain or discard information from the previous time step. The forget gate equation is as follows:

*ft*  (*Wxf*  *Xt* *Whf* *ht*1 *Wcf* *ct*1  *bf)*

(1)

1. Input gate: The input gate measures the weight of any fresh information that is being brought in. This is the input gate equation:

i*t* (*Wxi*  *Xt* *Whi* *ht*1 *Wci**ct*1  *bi)*

(2)

1. Output gate: These gates regulate information flow through the network and decide whether to recall or forget previous information. LSTM also has a memory cell that stores information over time, enabling the network to capture long-term dependencies in data. The output gate equation is as follows:

O*t*  (*WxO*  *Xt* *WhO* *ht*1 *WcO**ct*  *bO)*

(3)

The hidden state is calculated using the following equation:

*ht*  *ot* tanh(*ct*)

(4)

Utilising the forget gate's *ft* activation, the cell state is multiplied. The updated cell is multiplied by *ot* after being processed by the tan h function.

*ct*  *ftct*1  *it* tanh (*WxcXt* *Whcht*1  *bc*)

(5)

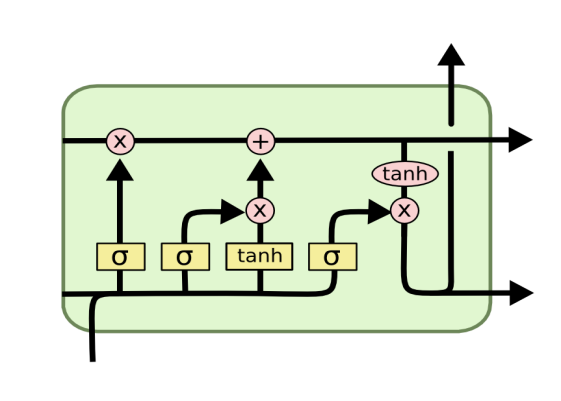
An LSTM model is often optimised using a loss function that gauges the difference between the projected and real stock prices in order to train it to forecast stock prices. The model can be trained using a variety of techniques, including back-propagation through time (BPTT) and gradient descent optimization. 

Figure 2: LSTM Architecture

## Bidirectional Long-Short-Term Memory (Bi-LSTM)

By processing the input sequence both forward and backward, the bidirectional LSTM architecture adds another level of complexity to the regular LSTM model. A forward computation is performed from instant 1 to moment t, and at each point, the forward hidden layer's output is retrieved and saved in the forward layer. The calculation was turned around at time 1 in order to extract and save the output of the backward hidden layer. Figure 3 demonstrates how this enables the network to record both past and future data, which may be especially helpful in time-series time-series prediction tasks, such as stock price prediction.

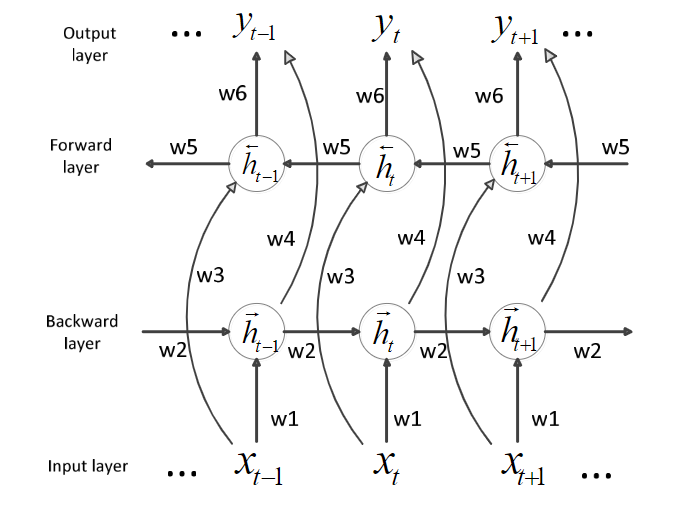


Figure 3: Bi-LSTM Architecture

The mathematical expressions for Bi-LSTM are as follows:

*ht =f(w1xt+ w2ht-1)*

(6)

*ht =f(w3xt+ w5ht+1)*

(7)

*ot=g(w4h4+ w6ht)*

(8)

## Gated Recurrent Unit (GRU)

Similar to an LSTM is a Recurrent Neural Network (RNN) architecture called a Gated Recurrent Unit (GRU). The GRU was introduced as a compressed version of the LSTM that can identify long-term associations in time-series data and is computationally less expensive and easier to train.

GRU features gating mechanisms that control the information flow through the network similarly to LSTM, allowing them to select whether data from previous time steps to keep or reject. The GRU, however, has fewer gating mechanisms than the LSTM, having just two gates

.

1. Gate update:How much of the previous data should be kept and forwarded to subsequent stages is decided by the update gate. Calculating the time step's *zt* requires:

*zt= σ(W(z)xt+U(z)ht-1)*

(9)

ii) Reset gate: The amount of previous information to be forgotten is decided by the reset gate.

*rt= σ(W(r)xt+U(r)ht-1)*

(9)

The reset gate is used to compute the memory content, which stores pertinent data from earlier phases.

*h́́t* *=tanh(Wxt+rt+‌⊙ Uht-1)*

(10)

The network uses the update gate to decide which data to collect from the previous phase *h́́t-1* in order to generate the *h́́t* vector for the current unit and which data to gather from the current memory contents.

*ht =zt‌⊙ ht-1+(1-zt)‌⊙ h́́t*

(11)

The framework for Gated Recurrent Unit (GRU) is given in figure 4:

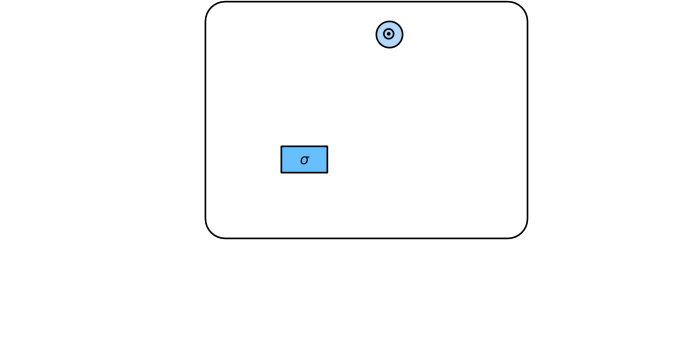


Figure 4: Gated Recurrent Unit(GRU)

## XGBoost

Extreme Gradient Boosting (XGBoost) is frequently employed for stock price forecasting. It is a boosting method that combines a number of weak learners (decision trees, for example) to produce an ensemble model that is robust.

In our suggested method, an XGBoost model is fine-tuned using pretraining to provide a regression prediction model for stock data. Through this procedure, we observed that the XGBoost fine-tuning model displayed superior generalization and predictive skills in comparison to other models. It can also manage nonlinear interactions between the input characteristics and the target variable, which is crucial for forecasting stock values that might display intricate and nonlinear patterns over time.

# **PROPOSED WORK**

1. DATA COLLECTION

Yahoo Finance was used to gather Apple stock data for 10 years, or from January 2013 to January 1, 2023. The Close, High, Low, and Opening Prices are included in the statistics. Before further training, these data were pre-processed and scaled between 0 and 1. Several deep learning techniques were used to perform Day Ahead Prediction.

1. DATA PRE-PROCESSING

For simplicity and ease of mathematical operation, the entire dataset for the time series was scaled between zero and one, which significantly improved the efficiency and training speed of the model. For a time period of 10 years, data worth 2518 days were collected, of which 1762 days (70%) were used for training and 756 days (30%) were used for testing. MinMax Scaler from the Sklearn library was used to perform this step. The normalization process is given by:

Anorm = (Aact － Amin )

(Amax － Amin)

1. PREDICTION ACCURACY MEASUREMENT

When evaluating the performance of machine learning models for stock price prediction, several metrics are commonly used to measure prediction accuracy. These include:

1. Mean Absolute Percentage Error (MAPE): The Mean Absolute Percentage Error (MAPE) determines the percentage difference between the expected and true figures.It's outlined as:

MAPE =(1/n) \* ∑(|actual - predicted| / actual) \* 100

where n is the sample size. When the magnitude of the mistakes is crucial, the MAPE is chosen for comparing the performance of several models.

1. Mean Absolute Error (MAE): Calculates the magnitude of the variation between expected and true values. It's outlined as:

MAE = (1/n) \* ∑(|actual - predicted|)

MAE is easy to interpret and useful when the magnitude of errors is important.

1. Root Mean Squared Error (RMSE):RMSE gives the sum of the squared discrepancies between the expected and observed values. It's outlined as:

RMSE =

RMSE is commonly used and is preferred when large errors are undesirable

IV)Coefficient of Determination(R-squared or R2): Determines the fraction of the volatility of the desired variable that can be forecast with the variables that are input. It's outlined as:

R2 = 1 - (SSres / SStot)

Where SSres is the sum of squared residuals.SStot is the total sum of squares. R2 lies between 0 to 1, with higher values indicating better predictive accuracy.

# **Result analysis**

The experiment was performed with an Intel Core i7 (1.30 GHz) processor and 16GB RAM. The proposed model was built in Python on the Visual Studio Code IDE using TensorFlow. The required graphs were plotted using the matplotlib library. A Moving Average Technical Indicator was considered for this study. Figure 5 shows the original closing price for apples over 10 years, and Figure 6 shows the relationship between the Moving Average of closing prices over 200 and 100 days with respect to the closing price.



Figure 5: Closing Price of Apple (2013-2023)

The graph is non linear and has several highs and lows and thus requires a good technique for prediction.The table below shows the performance of various RNN techniques to predict the closing price over the period of 10 years.

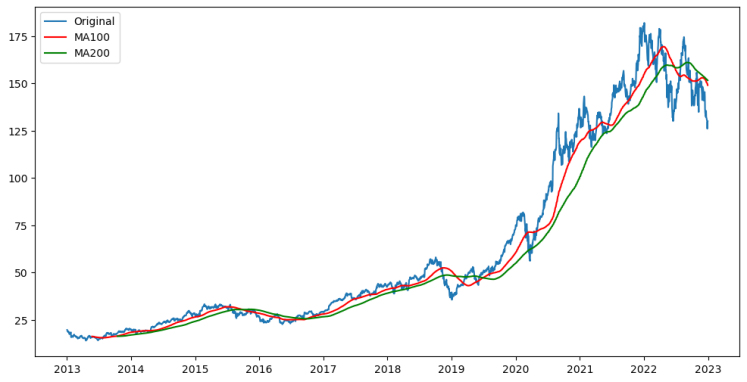


Figure 6: Moving Average(100 days) and Moving Average (200 days) with respect to the Original closing price.

The moving average for 100 days moves closely with the closing price of the stock, and was hence considered for this study.

Table 1 shows the study of the different prediction models for the Apple data.

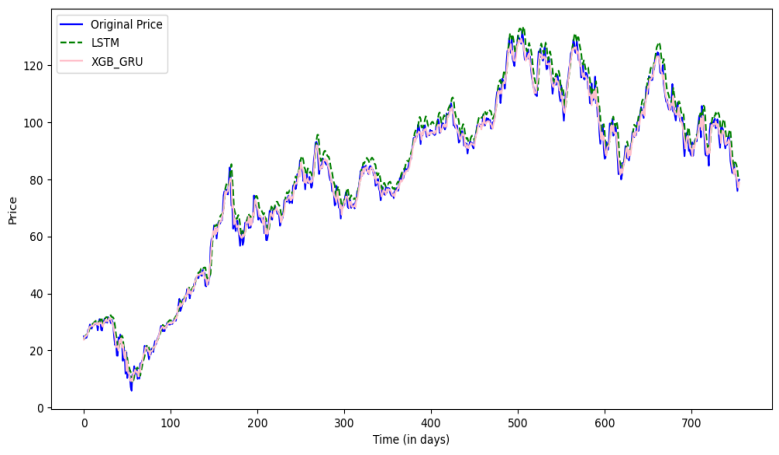
(MAPE,MAE,RMSE in %)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Technique** | **MAPE** | **MAE** | **RMSE** | **R2** |
| LSTM | 4.714 | 3.036 | 3.914 | 0.9835 |
| BI-LSTM | 3.854 | 2.349 | 3.066 | 0.9899 |
| GRU | 3.440 | 2.090 | 2.806 | 0.9915 |
| XGBOOST BILSTM | 2.970 | 1.780 | 2.333 | 0.9941 |
| **Proposed Method** | **2.856** | **1.682** | **2.234** | **0.9946** |

Table 1: Comparison of Performance of the different model

Our proposed approach, which combines the features of GRU and feeds them to XGBoost, outperforms all other prediction techniques for the given dataset. This method can also be applied to other datasets and to stock tickers. The results in Table 1 demonstrate that our proposed method achieved the lowest error compared with the other methods.

Figure 7 shows the comparative study between the proposed model and ordinary LSTM model.

 Figure 7: Assessment of the performance of the models, a comparison between the initial stock prices and the forecasts produced by the LSTM model and the suggested hybrid model is made.

This analysis provides insights into the accuracy of the models in predicting stock prices and their effectiveness in capturing the unpredictable factors that influence stock prices.

Upon observation, it can be noted that the proposed model exhibits greater accuracy and proximity to the closing price in comparison to the LSTM model.

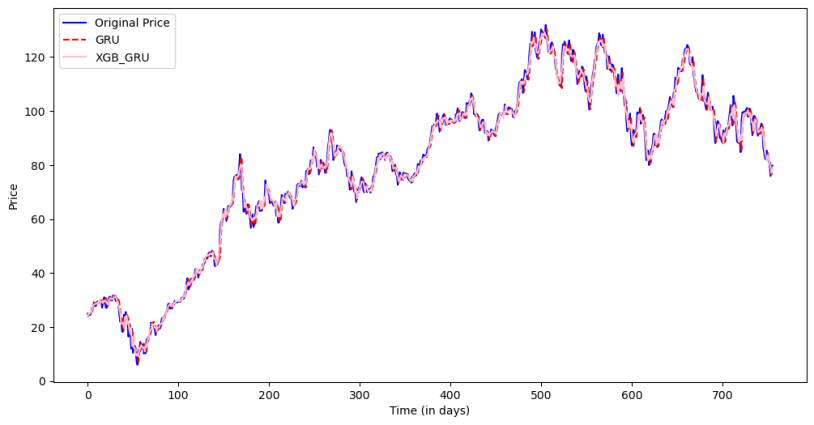


Figure 8: Comparisons between Original price against the predictions made by GRU and the proposed model.

# **Conclusion and future work**

Overall, this study aims to provide a detailed analysis of three widely used machine learning algorithms and two hybrid models for predicting the stock market. The results of this study are valuable to financial analysts, traders, and investors. However, it should be noted that stock prices are affected by various unforeseeable factors like global events, economic conditions, and market sentiment, which may cause unexpected fluctuations in prices. Therefore, it is essential to consider all of these factors and employ various techniques to accurately analyze and predict stock prices.

##### References:

1. Nirupama Parida, Bunil Kumar Balabantaray, Rajashree Nayak, Jitendra Kumar Rout. "A Deep Learning based Approach to Stock Market Price Prediction using Technical indicators", 2023 International Conference on Advances in Intelligent Computing and Applications (AICAPS), 2023
2. CHEN Kai, ZHOU Yi, DAI Fangyan. A LSTM-based method for stock returns prediction: A case study of China stock market[C] //IEEE International Conference on Big Data. Santa Clara:IEEE Press,2015:2823-2824. DOI :10.1109/BigData.2015.7364089..
3. Mingzhu Jia,Jian Huang,Lihua Pang, Qian Zhao 'Analysis and Research on Stock Price of LSTM and Bidirectional LSTM Neural Network'3rd International Conference on Computer Engineering, Information Science & Application Technology (ICCIA 2019)
4. Weng, Bin, et al. “Prediciting short-term stock prices using ensemble methods and online data soruces.” Expert Systems with Applications 112(2018) : 258-273
5. T. Chen and C. Guestrin, “Xgboost: A scalable tree boosting system,”in KDD, 2016.
6. Naeini, Mahdi Pakdaman, Hamidreza Taremian, and Homa Baradaran Hashemi. "Stock market value prediction using neural networks." 2010 international conference on computer information systems and industrial management applications (CISIM). IEEE, 2010.