Developing a Web Application to Predict Stock Price in Bangladesh Using Machine Learning

Project Proposal

Supervisor

Priyam Chowdhury

Submitted by

Md. Moinuddin Kamal

ID: 006-40-08

Department of Computer Science and Engineering

Southern University Bangladesh

Date of Submission: 05-06-2023

Southern University Bangladesh

Department of Computer Science & Engineering

Project Proposal

Application for the Approval of B.Sc. Engineering Thesis/Project

Tentative Title: **Developing a Web Application to Predict Stock Price in Bangladesh Using Machine Learning**

Student Name : Md. Moinuddin Kamal Session : 2022-2023

ID : 006-40-08

: Batch: 40

Supervisor Name : Priyam Chowdhury

Designation : Lecturer

Department of Computer Science & Engineering

Department : Computer Science & Engineering

Program : B.Sc. Engineering

Table of Contents

1. Introduction

Basically, large-capitalized quantitative traders acquire stock derivatives and equities at low prices and then sell them at high prices. Due to the size of the markets and the speed at which deals are conducted, investors used to rely on their own experience to spot market trends, but this is no longer practical. Even though the tendency in stock market forecasting is nothing new, numerous organizations continue to discuss it. Before purchasing a stock, investors do two different types of stock analysis. The first type is called fundamental analysis; during this research, investors consider the intrinsic worth of the stock as well as the performance of the market, the economy, and the political environment. On the other side, technical analysis tracks the development of stocks by the examination of statistics produced by market activity, such as past prices and volumes. Due to its volatile nature the fluctuations of price are unpredictable most of the times, for which the investor has to face severe loss [1]. Numerous optimization strategies that were inspired by nature have recently been created and successfully used in numerous Financial Engineering domains. Predicting stock prices is a cumbersome task as it does not follow any specific pattern. Changes in the stock prices are purely based on supply and demand during a period of time. In order to learn the specific characteristics of a stock price, we can use algorithm to identify these patterns through machine learning. One of the most well-known networks for series forecasting is LSTM (long short-term memory) which is a Recurrent Neural Network (RNN) that is able to remember information over a long period of time, thus making them extremely useful for predicting stock prices. RNNs are well-suited to time series data and they are able to process the data step-by-step, maintaining an internal state where they cache the information they have seen so far in a summarized version. The successful prediction of a stock's future price could yield a significant profit

2. Background and Literature Review

A stock is a type of financial security that represents the ownership, or equity interest, of a fraction of a corporation. That equity is established on a per share basis, and the owners are often referred to as shareholders or stockholders. Thus, when you buy a share — or multiple shares — of stock, you are purchasing a proportionate claim on a company's net assets and future earnings.

in many cases, there are a few machine learning techniques which combine the broader categories of specialized analysis with fundamental analysis approaches to anticipate the stock markets. Figure 2.1 appears as a scientific categorization of well known stock prediction methods. These methods have picked up popularity and have appeared promising results in the field of stock investigation. Section 5 discusses in detail on the various methodologies used by researchers.

Contributing in stocks can be a key portion of your individual fund methodology. The essential reason most individuals purchase stocks is to produce a long-term return on their investment (ROI) that surpasses that of other conspicuous resource classes, such as bonds, genuine domain and commodities. ROI is a normally used profitability ratio that measures the quantity of return, or profit, and funding that generates relative to its costs. ROI is expressed as a percent and is extraordinarily beneficial in comparing character investments or competing funding opportunities. A ”great” ROI depends on a few components. The foremost critical thought in deciding a great ROI is your budgetary requirement. Most financial specialists would see a normal yearly rate of return of 10% or more as a great ROI for long-term speculations within the stock showcase.

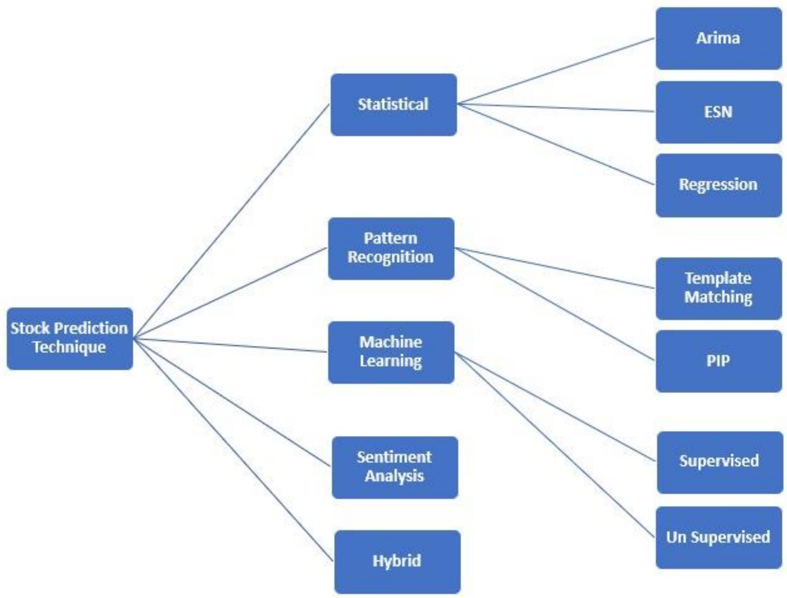

Figure 2.1: Taxonomy of Stock Prediction Techniques

2.1 Previous literature:

Various algorithms for machine learning are used to predict stock price trends. Some of them are ANN(Artificial Neural Networks) [3,4,5,6] , GA(Genetic Algorithm) [5], LS-SVM(Least Square Support Vector Machine) [1,7], Trend Estimation with Linear Regression (TELR) [8],SVM(support vector machines) [4,6,9] with different kernels, KNN(K Nearest Neighbors) [5] structural support vector machines (SSVMs)[10] . Some statistical analyses are also used like Autoregressive Integrated Moving Average (ARIMA) [11]. But none of them were able to give quite promising prediction due to the non-linearity of the data.

2.2 Algorithm performance of different models on Stock Price Data

Studies have tried to predict stock prices using Artificial Neural Network. In a study in 2017 [4] on Korean stock market ANN was used to predict the stock price with the highest accuracy of 81.34 percent for 20 days and 83.01 percent for 30 days moving average. In another study [6] ANN

was used and got 86.69% average accuracy based on experiment carried out on three different stocks. In 2018 this study [3] ANN was used and the best SSE score was 0.6271104815 for Apple, 0.0121281374 for Pepsi, 0.0335425612 for IBM, 0.016770174 for McDonald and 0.0211154625 for LG. In this study [4] author used ANN and achieved 96.10 percent accuracy for 5-fold average prediction (Mj`). Mj is a model where $1 \leq j \leq J$, for each classifier. And Mj` is a variant model as a substitution of Mj. 92.81 percent accuracy was achieved from model Mj.
Least square support vector machine (LS-SVM) is another popular machine learning used to predict stock prices. In this study [7] author a LS-SVM model along with Particle Swarm

Optimization (PSO) and the accuracy rate of the system was around 90.5 – 93%. Another research [1] used LS-SVM with PSO optimization and LS_SVM to predict stock prices. The Mean Squared Error (MSE) was 0.5317 for Adobe, 0.6314 for oracle 0.7725 for HP and 0.7905 for American Express with PSO-LS-SVM where MSE was 0.5703 for Adobe, 0.8829 for Oracle 1.2537 for HP and 1.0663 for American Express.

Other potent Machine Learning Algorithms like SVM was used in this study [6] and found average accuracy of 89.33 percent after applying the algorithm for three company stocks. For BSE-Sensex the accuracy was 90.10% using polynomial kernel where c=1 and degree =1 and for RBF kernel the accuracy was 88.08 where c=1 gamma=4. For Infosys the accuracy was 89.59% with polynomial kernel where c=1 and degree=1 and 87.80% for RBF kernel where c=5 and gamma=1.5. Another research [9] that used SVM with RBF kernel has the mean accuracy of 53.3% to 56.8% for 10 days. And the accuracy rate can be lower than 50% for different dataset. A modified linear regression algorithm Trend Estimation with Linear Regression was used in this study [8] and Mean Absolute Percentage Error (MAPE) was 5.41% for bank data and 5.42% for overall stock data. K Nearest Neighbors or KNN is also been used to predict stock prices. A study that used KNN [5] has an accuracy of 83.52 percent with KNN. Structural support vector machines (SSVMs) is used in a research and the accuracy with training samples was higher than 78% and the accuracy with testing samples was about 50%.

2.3 Research gap:

None of the algorithm we have seen to use in those studies that we have discussed had an accuracy higher than 90 percent except SVM optimized with PSO [7]. In that research a highest accuracy of 90.5 to 93 percent was found. One study was found which has used Bangladeshi stock price data to predict stock price. Author has used a modified linear regression algorithm named TELR. No study was carried out on Bangladeshi Stock market data with a simple linear regression algorithm and SVR with different kernels. So we have implemented the model which will use simple linear regression and SVR with RBF and linear kernel and then compare the accuracy among those.

3. Objective

Over the previous few decades, many social science researches have focused on predicting social and financial development tendencies with quantitative methods. Many viable techniques in time-series analysis, each with benefits and disadvantages, can be interpreted as methods for using past facts to construct forecasts and strategies on future value. Based on all of this, this paper takes data to forecast the future stock price through forecast models so as to formulate the most advantageous funding approach for investors to refer to at a positive extent. For this motive we have develop web based application used python scripting language , Django Framework ,JavaScript, HTML, CSS which has a speedy execution environment and this will assist out the buyers in order to make a prediction on what shares money have to be invested, it will also help in maintaining the most economical stability of the share market. Future work can be completed with the aid of running these python script code with greater superior functions.

- To Make a comparative study of deferent of machine learning algorithms to predict the future stock price of companies , starting with simple algorithms like Random forest, and then move on to advanced techniques like ARIMA, LSTM, bidirectional LSTM.
- To develop a Web Application of predicting of future stock price. Interactive User Interface for Chart, Prediction Search Menu, Analysis Report. User and admin login UI .

4. Problem Description:

Most investors put their money into stocks based on intuition or pure guesswork, hoping that the price will go up and they make a profit. It doesn't happen and most of the time you face a loss. Losses can be minimized if you know how to use the right forecasting strategy correctly. Stock market forecasts help investors achieve consistency as traders as they aim to make more money than they lose in this volatile market. Applying the correct stock prediction technique will help investors know better about entry and exit points. So often the traders either enter or exit the market at the wrong time which means they fail to capitalize on the full potential of making profits. These days more and more critical information about the stock market has become available on the Web [3]. Therefore, people are getting influenced by others' opinions and information and making decisions accordingly. Nowadays, advanced intelligent techniques based on either technical or fundamental analysis are used for predicting stock prices [4]. Hence, the upcoming stock price prediction system will help investors in decision making for which company's stock to invest into and that it will eventually help them gain more profit and refrain them from choosing companies which will not be feasible for them to invest in.

5. Dataset Preparation

Building the information set is one of the most challenging tasks of any machine learning project. The struggle gets even scarier when we don't have any benchmark information set for our issues. Presently, building a data set can be exhausting in some diverse ways. I will briefly examine the ins and outs of each technique and our selections among them.

5.1 Data Collection:

There are huge depots of datasets collected from stock exchanges all over the world. As mentioned before, in our country there are two stock exchanges, DSE and CSE. And, we have selected to do our analysis on DSE only. The website contains data archives of more than 600 trading companies participating in the stock market. Figure 4.1 shows a glimpse of the dataset that is available on the DSE official website.

The description of the attributes included in this dataset are as follows:

- VOLUME : It is the number of shares of a security traded during a given period of time.
- LOW : The lowest price that a stock trades on that day.
- OPENP* : Opening Price
- CLOSEP* : Closing Price
- YCP : Yesterday's Opening Price
- TRADE : Name of the stock trading company
- VALUE (mn) : shares of a company
- LTP* : Last Traded Price
- HIGH : The highest price at which a stock traded during the course of the trading day

#	DATE	TRADING CODE	LTP*	HIGH	LOW	OPENP*	CLOSEP*	YCP	TRADE	VALUE (mn)	VOLUME
1	2022-08-04	1JANATAMF	6.4	6.4	6.3	6.3	6.4	6.3	83	2.099	330,390
2	2022-08-03	1JANATAMF	6.3	6.4	6.3	6.3	6.3	6.3	74	1.518	239,091
3	2022-08-02	1JANATAMF	6.3	6.4	6.2	6.3	6.3	6.2	121	2.766	439,037
4	2022-08-01	1JANATAMF	6.2	6.3	6.2	6.3	6.2	6.3	73	1.447	232,248
5	2022-08-04	1STPRIMFMF	18	18.3	17.8	18	18	18.2	74	3.201	177,346
6	2022-08-03	1STPRIMFMF	18.1	18.3	17.7	17.7	18.2	17.6	313	11.436	632,223
7	2022-08-02	1STPRIMFMF	17.6	17.9	17.6	17.6	17.6	17.4	215	7.315	412,974
8	2022-08-01	1STPRIMFMF	17.6	17.7	17.3	17.4	17.4	17.8	128	2.999	171,770
9	2022-08-04	AAMRANET	38.5	38.8	38	38.8	38.5	38.6	813	39.129	1,018,925

Figure 4.1: Data Archive from Dhaka Stock Exchange (DSE)

5.2 Scraping the web:

Now that we have data available on the website, it will be cumbersome to individually copy and paste the row wise data from the website directly. In order to save time and energy, we can attempt this step to load a huge amount of data for our data set. BeautifulSoup library is used to extract content from an HTML page. Three basic steps are required to be followed in order to undergo the web scraping process. At first using the requests library, we have to extract the HTML content. After analyzing the HTML structure and identifying the tags which have our content, we then extract the tags using Beautiful Soup and put the data in a Python list. That is how we obtained our dataset.

5.3 Data Pre Processing:

Before feeding the dataset, it requires a few pre-processing steps. Firstly, we had to convert the date column within the dataset to date time format. Then, we created a separate dataframe where we removed the column containing the trading codes. Here the trading codes are the companies that are involved in the stock trading. We also had to clean the separated dataframe that didn't contain the company name column. We removed all commas and spaces in between the values within the dataset. From our list, we then selected five stock companies with which we are going to implement this project. Then, all prices that were prior to 2022 were considered as our training set and the rest of that data has been used as the test set.

We rescaled all stock prices to zero for the lowest and 1 for the highest. Each company has their own scale. We made another two dictionaries which contain scaled prices for each company. One contains a train set and another contains a test set. We also created another dictionary for collecting the scaler. This will be useful when we want to inverse transform our prediction. Finally, our training and test sets are ready!

6. Methodology

6.1 Process Flow

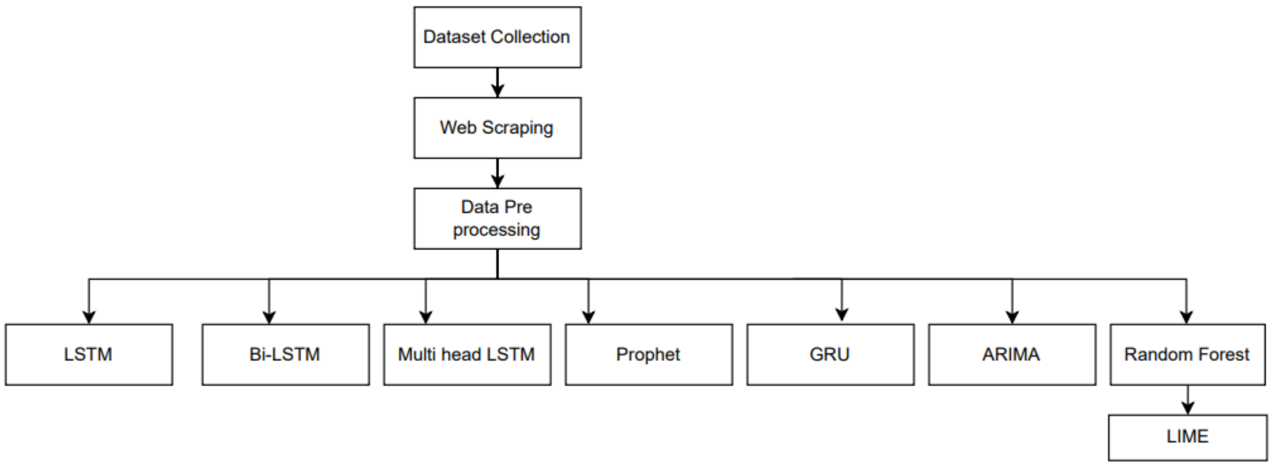

Figure 5.1: Work flow Diagram

In this section, we present some the proposed methods and the design of the proposed solution…

6.2 LSTM:

Standard Recurrent Neural Networks (RNNs) are affected by short-time period memory because of a vanishing gradient trouble that emerges whilst operating with longer information sequences. Luckily, we’ve got greater superior variations of RNNs that could maintain vital statistics from in advance elements of the series and convey it forward. The best-regarded variations are Long Short-Term Memory (LSTM) and Gated Recurrent Units (GRU). It addressed the difficulty of RNN long-time period dependency, wherein the RNN is unable to expect words saved in long-time period reminiscence however could make extra correct predictions primarily based totally on contemporary statistics. RNN now no longer offers a green overall performance as the space duration rises. The LSTM may also hold data for a long term through default. It is used for time-collection statistics processing, prediction, and classification.

Let me begin with a brief recap of a simple RNN structure. RNN includes more than one layer just like a Feed-Forward Neural Network: the enter layer, hidden layer(s), and output layer as shown in Figure 3. However, RNN contains recurrent units in its hidden layer, which lets in the set of rules to process series data. It does it with the aid of frequently passing a hidden state from a previous time step and mixing it with an input of the cutting-edge one. However, RNN contains recurrent units in its hidden layer, which lets in the set of rules to process collection data. It does it via means of commonly passing a hidden state from a previous time step and mixing it with an entry of the present day one

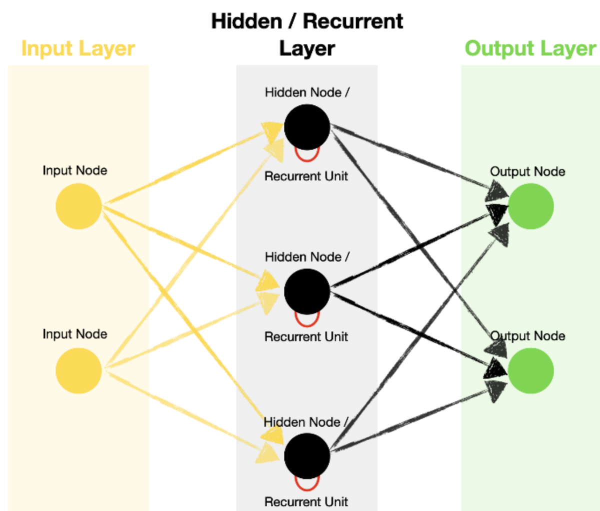

Figure 5.2: Standard Recurrent Neural Network architecture

We realize that RNNs utilize recurrent units to research from the series data. So do LSTMs. However, what occurs within the recurrent unit could be very special among the two. Looking within the simplified recurrent unit diagram of a fashionable RNN (weights and biases now no longer shown), we observe that there are simplest important operations: combining the preceding hidden kingdom with the brand new enter and passing it through the activation function. After the hidden state is calculated at timestep t, it is surpassed, returned to the recurrent unit and mixed with the enter at timestep t+1 to calculate the brand new hidden state at timestep t+1. This method repeats for t+2, t+3, t+n till the predefined number (n) of timesteps is reached. Meanwhile, LSTM employs numerous gates to determine what data to maintain or discard. Also, it provides a 13 cell state, which is sort of a long-time period reminiscence of LSTM. The LSTM architecture consists of a set of recurrently connected sub-networks, known as memory blocks [23] . The idea behind the memory block is to maintain its state over time and regulate the information flow through non-linear gating units. Four neural networks and a large number of memory cells, which are arranged in a chain pattern, make up the LSTM as shown in Figure 5.1. A cell, an input gate, an output gate, and a forget gate make up a typical LSTM unit. Three gates regulate the information flow into and out of the cell, and the cell retains values for arbitrary time periods. Time series with indeterminate duration can be categorized, examined, and predicted with the LSTM algorithm

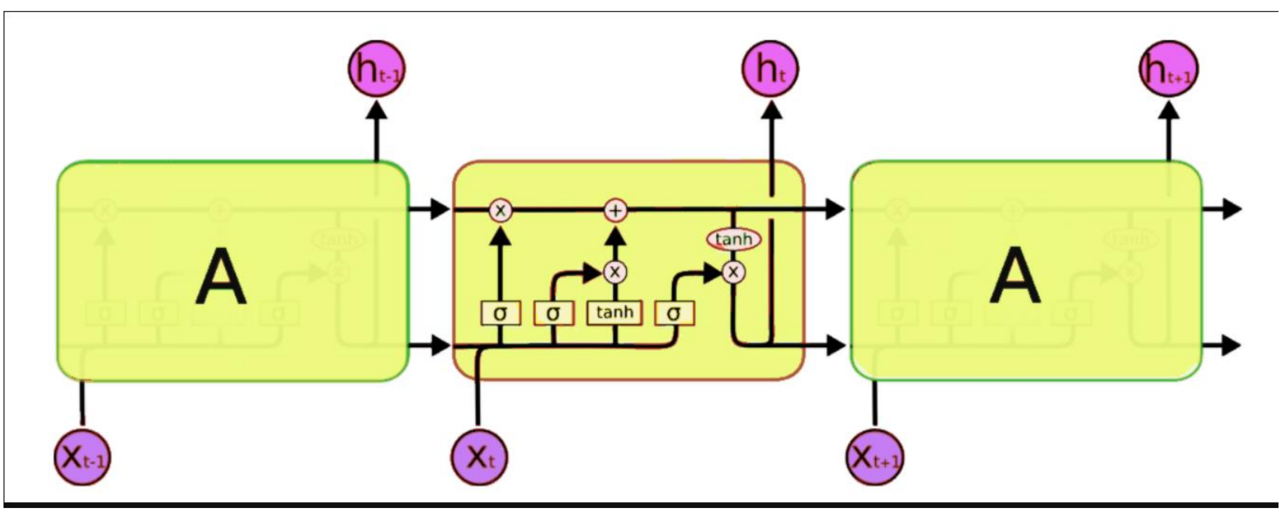

Figure 5.3: Architecture of LSTM

There are three entrances:

Input Gate: It determines which of the input values should be used to change the memory. The sigmoid function determines whether to allow 0 or 1 values through. And the tanh function assigns weight to the data provided, determining their importance on a scale of -1 to 1.

$i_t = \sigma(W_i.[h_t - 1, x_t] + b_i)$ (5.1)

$C_t = tanh(W_c.[h_t - 1, x_t] + b_c)$ (5.2)

Forget Gate: It fnds the details that should be removed from the block. It is decided by a sigmoid function. For each number in the cell state C_t - 1, it looks at the preceding state (h_t - 1) and the content input (X_t) and produces a number between 0 and 1.

$f_t = \sigma(W_f.[h_t - 1, x_t] + b_f)$ (5.3)

Output Gate: The block's input and memory are used to determine the output. The sigmoid function determines whether to allow 0 or 1 values through. And the tanh function determines which values are allowed to pass through 0, 1. And the tanh function assigns weight to the values provided, determining their relevance on a scale of -1 to 1 and multiplying it with the sigmoid output

$O_t = \sigma(W_o.[h_t - 1, x_t] + b_o)$ (5.4)

$h_t = o_t * tanh(C_t)$ (5.5

6.3 Bidirectional LSTMs

An improvement on LSTMs that is frequently considered is bidirectional LSTMs. Bidirectional Recurrent Neural Networks show each training sequence both forward

and backward to two separate recurrent nets that are coupled to the same output layer (BRNN). In other words, the BRNN is fully sequentially aware of every point before and after every point in a given sequence. Additionally, since the internet is free to use as much or as little of this context as it needs, there is no need to provide a (task-dependent) time window or target delay size.
The flow of information from the backward and forward layers is depicted in the diagram (Figure 5.3). BI-LSTM is typically used when activities requiring sequence to sequence are required. Speech recognition, text categorization, and forecasting models can all employ this type of network.

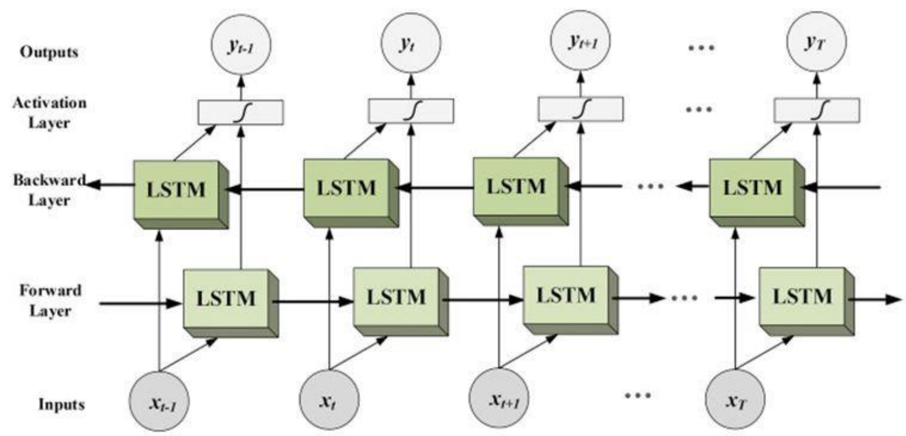

Figure 5.4: Bi-LSTM Neural Network Structure

6.4 Multithreaded Attention Based LSTM

The multi-head attention scheme was frst put forth by Vaswani et al. [24]. Their research discovered that it is advantageous to use multi-head attention for the queries, values, and keys by using an attention layer as a function, which maps a query and a set of key-value pairs to the output. The multi-head attention layer computes the hidden information by linearly projecting the context vectors into several subspaces, performing better than single-head attention. We calculate the output using weighted values, which are determined by queries and the related keys, as inspired by Vaswani et al. (2017) [24].

For attention based, we simply employ two LSTM output varieties. The fact that the output of all time comprises data from every LSTM output makes it crucial. The last time step output was chosen since it has the most redundant data of all the time steps.

The multi-head attention scores and context vectors are calculated as follows:

$$si = softmax(Qi \times KiH),\ si \in RB,1,T \quad (5.6)$$

$$contexti = si \times Vi,\ contexti \in RB,1,Zn \quad (5.7)$$

$$CV = Concat([context1, ..., contextn]),\ CV \in RB,1,Z \quad (5.8)$$

where *si* represents the multi-head time-dimension attention score and *context* represents the reduced-dimension context vectors from each subspace.

7. Implementation and result analysis:

In this section, we basically will demonstrate the performance of our stock prediction system to effectively detect and predict our predefined closing price of five companies. Setup starts by importing all necessary libraries (NumPy), (pandas), (matplotlib). Load the dataset and define the target variable for the problem. Then import the CSV file into Python using read csv() from pandas. The dataset is of the following from the table shown in Figure 7.1:

7.1 Implementation Details

	DATE	TRADING CODE	LTP*	HIGH	LOW	OPENP*	CLOSEP*	YCP	TRADE	VALUE (mn)	VOLUME
0	9/30/2020	1JANATAMF	5.9	5.9	5.4	5.4	5.9	5.4	611	40.598	7,146,732
1	9/29/2020	1JANATAMF	5.4	5.5	5.2	5.2	5.4	5.2	328	17.077	3,166,516
2	9/28/2020	1JANATAMF	5.2	5.4	5.1	5.1	5.2	5.1	320	19.421	3,690,552
3	9/27/2020	1JANATAMF	5.1	5.3	5	5.2	5.1	5.2	278	16.453	3,214,513
4	9/24/2020	1JANATAMF	5.3	5.5	5.1	5.4	5.2	5.3	333	15.498	2,917,450

Figure 7.1: Snippet of the Dataframe

After rigorous data processing as explained in section 4, there is a need to extract the feature which is required for data analysis, then divide it as testing and training data, training the algorithms to predict the price and the final step is to visualize the data. First, we trained our LSTM model using the training dataset which as mentioned before contained data prior to 2022 as a testing test for five selected companies.

A cell, an information door, an entrance door, and a door with a view make up a standard LSTM unit. The three inputs control how quickly data centers and leaves the cell as the cell collects values over arbitrary time intervals. The LSTM's ability to learn context-specific temporal dependency is its key benefit. Without explicitly applying the activation function within the recurrent components, each LSTM unit gathers data for either a lengthy or brief duration (thus the name). It is important to keep in mind that every cell state is significantly amplified by the output of the ignored entrance, which fluctuates between 0 and 1. Feature scaling on the dataset is done so that the data values vary from 0 and 1.

The RNN (Recurrent neural network) is then built for the data set and the RNN is initialized by using a sequential repressor. The first LSTM layer is added and the second, third and fourth LSTM layer is added with some dropout regularization for removing the unwanted values. Also, then the output layer is added. Lastly, compiling the RNN by adding RMSprop Optimizer and the loss as mean squared error, mean absolute percentage error, mean absolute error and r2 score. The final step is to plot the data using a visualization technique that helps to show the variation of data in the outcome of our algorithm error, mean absolute error and r2 score. The final step is to plot the data using a visualization technique that helps to show the variation of data in the outcome of our algorithm.

We can improve our prediction by introducing shifting/lagging. Essentially, we slide our prediction for a period of time. This is a common practice in the signal processing subfield. When we try to make our prediction start earlier, we call it lagging. As for consequences, lagged prediction will last – equal to how much we displace the prediction – value equal to NaN. If we lag it by 5 day, then the last 5 day prediction will become NaN. When we displace to make our prediction start later, we call it shifting. As for consequences, shifted prediction will have first – equal to how much we displace the prediction – value equal to NaN. If we shift it by 5 day, then the last 5 day prediction will become NaN.

Similarly the Gated Recurrent Units Model was implemented using the similar procedure as explained for the LSTM Model but the optimizer that was used was SGD optimizer. However, the current version of GRU uses a dense GRU network with 100 units as opposed to the GRU network with 50 units in the previous version. The biLSTM model was implemented as well following the same process as the LSTM Model using the RMSprop Optimizer and also the multihead LSTM was used with sigmoid activation function.

After fitting these models, we now move to the Prophet model. To use Prophet for forecasting, first, a Prophet() object is defend and configured, then it is fit on the dataset by calling the fit() function and passing the data. The Prophet() object takes arguments to configure the type of model we want, such as the type of growth, the type of seasonality, and more. By default, the model will work hard to figure out almost everything automatically. We have checked yearly seasonality to be true here. Then finally, we fit the model with the training set and predict using the test set. Now comes the last model fit to our dataset and it is the ARIMA model. The stats models library provides the capability to fit an ARIMA model. At first we define the model by calling ARIMA() and passing in the p, d, and p parameters. The model is prepared on the training data by calling the fit() function and here we fit an ARIMA(1,1,0) model. This sets the lag value to 1 for auto regression, uses a difference order of 1 to make the time series stationary, and uses a moving average model of 0. The predictions can be made by calling the predict() function and specifying the index of the time or times to be predicted. Furthermore, we used our dataset to ft. a random forest regression model. Random forest means data about data estimators. It fits a number of decision trees on various sub samples of the given data. It controls over-fitting and improves predictive accuracy.

From the dataset pick N random records, based on N records, build a decision tree. We have chosen the number of trees to be and repeat steps 1 and 2. In case of a regression problem, for a new record, each tree in the forest predicts a value for Y (output).

Now that we have trained and predicted our models with our dataset, our aim is to comprehend why a specific prediction was made by the machine learning model. When different versions of our data are fed into the machine learning model, LIME examines what happens to the predictions. LIME creates a brand-new dataset made up of altered samples and the related black box model predictions. We used the LIME Model Interpreter based on Random Forest Regressor for the target label as “Close Price”

Furthermore, we will discuss the outcomes of these models in the Results section.

7.2 Result Analysis

The suggested LSTM model is implemented in Python and uses past data to forecast the price of BPML, ASIAINS, SAIHAMCOT, ARAMITCEM and BEXIMCO shares in the future. The visualization of all these company forecasts is shown in the images below. The graph below which is Figure 6.2, from our algorithm will display the expected price of all the five company shares in our paper, which implements an algorithm that predicts the stock price of a share for a specific length of time. Plotted from the output of our algorithm using LSTM units to achieve accuracy is the result shown in the graph below.

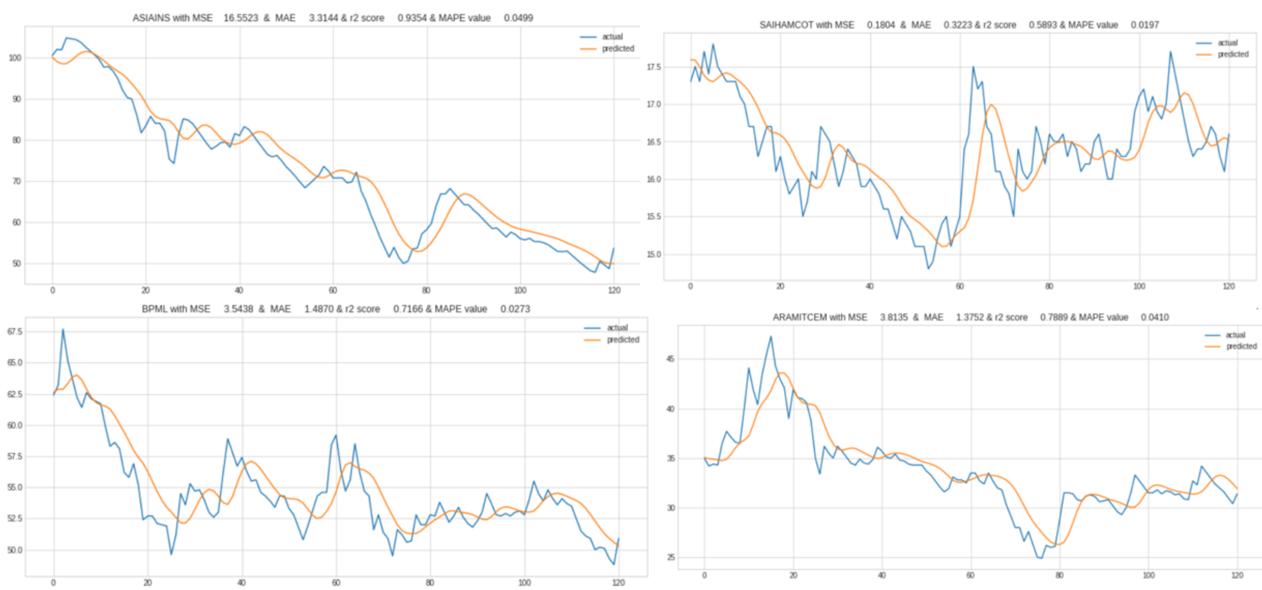

Figure 7.2: Results from LSTM Model

Figure 7.3: Results from LSTM Model on BEXIMCO

8 Tools/Technology

Stock Market Prediction Web based project need a lot of library and programing language which use for imported data analysis.
In this project I am using Python programing language and Django framework for web base project perform.
Language: Backend
Python
Framework: Django

Frontend: HTML, CSS, Bootstrap, JavaScript,
Data analysis Library:

- **Pandas** – This library helps to load the data frame in a 2D array format and has multiple functions to perform analysis tasks in one go.
- **Numpy** – Numpy arrays are very fast and can perform large computations in a very short time.
- **Matplotlib**/**Seaborn** – This library is used to draw visualizations.

- Sklearn – This module contains multiple libraries having pre-implemented functions to perform tasks from data preprocessing to model development and evaluation.
- **XGBoost** – This contains the eXtreme Gradient Boosting machine learning algorithm which is one of the algorithms which helps us to achieve high accuracy on predictions.

8.1 Budget Estimation:

In this section to estimation potential budget for implements this project

Sl	Description	Quantity	Price
1	Laptop i7 12gth gen with SSD, mouse ,keyboard	1nos	85000
2	Domain and hosting	1 Nos	3000
3	Report ,printing cost	1 nos	2000
4	Entertainment	No specific	5000
5	Consultant fee (Data analysis and market survey)	No specific	15000
		Total	110000

So approximately need **110000 (one lakhs teen thousands)** for prepared this project

9. Recommendations for Future Work

We leveraged the combinations of price, volumes and corporate statistics as input data. We proposed, developed, trained and tested four models: LSTM, Bi-LSTM, Multihead LSTM, ARIMA, and Prophet Models, and built up Long-Only and Long-Short trading strategies according to our model predictions. The research used DSE stocks historical data for the past years from September, 2020, to July 2022, to compare the multi models' results. The ARIMA model shows more superior results over other models due its ability to assign different weights to the input features hence automatically choose the most relevant features. Hence the ARIMA model is more able to capture the long-term dependence in the time series and more suitable in predicting financial time series. Our superior trading return from ARIMA model further validates our experimental result. This could guide investors in stock market to make profitable investment decisions. With the results obtained ARIMA models can compete reasonably well with emerging forecasting techniques in short-term prediction. From the analysis the different investors can choose companies according to their returns. my study did have some drawbacks, though. The range of data accessible was not the same for all companies because different companies were listed for stock trading at various times.

We were unable to detect correlations between the companies whose data varied greatly because of this discrepancy. As a result, we had to exclude companies from the study whose data did not begin at the time we wanted and that is what resulted us in choosing four companies to work with. Additionally, a sizable quantity of data was missing from some of the corporate datasets we chose. The dataset's prior values have to be used to fill in the missing values. If the data had been more reliable, we could have further improved our model. One direction of future work will be dealing with the volatility of stock time series. One difficulty of predicting stock market arises from its non-stationary behaviour.It would be interesting to see how ARIMA performs on demised data. Moreover, in future more fine tuning of the models can provide us more accurate results in the future. Presently this project (Stock Market Price Prediction) made by python and Django framework .so its need to move in future to build API (Application Programmable Interface) for real time access web platform and mobile platform.

References

1.Hegazy, O., Soliman, O. S., & Salam, M. A. (2014). A machine learning model for stock market prediction. arXiv preprint arXiv:1402.7351,

2. Guo, K., Sun, Y., & Qian, X. (2017). Can investor sentiment be used to predict the stock price? Dynamic analysis based on China stock market. Physica A: Statistical Mechanics and its Applications, 469, 390-396.

3. Ebadati, O. M. E., & Mortazavi, M. T. (2018). An efficient hybrid machine learning method for time series stock market forecasting. Neural Network World, 28(1), 41-55.

4. Qian, B., & Rasheed, K. (2007). Stock market prediction with multiple classifiers. Applied Intelligence, 26(1), 25-33.

5. Pyo, S., Lee, J., Cha, M., & Jang, H. (2017). Predictability of machine learning techniques to forecast the trends of market index prices: Hypothesis testing for the Korean stock markets. PloS one, 12(11), e0188107.

6. Patel, J., Shah, S., Thakkar, P., & Kotecha, K. (2015). Predicting stock and stock price index movement using trend deterministic data preparation and machine learning techniques. Expert Systems with Applications, 42(1), 259-268.

7. Efat, M. I. A., Bashar, R., Uddin, K. I., & Bhuiyan,(2018) T. Trend Estimation of Stock Market: An Intelligent Decision System. International Conference on Cyber Security and Computer Science.

8. Madge, S., & Bhatt, S. (2015). Predicting stock price direction using support vector machines. Independent work report spring.

9. Leung, C. K. S., MacKinnon, R. K., & Wang, Y. (2014, July). A machine learning approach for stock price prediction. In Proceedings of the 18th International Database Engineering & Applications Symposium (pp. 274-277). ACM.

10. Huang, C., Huang, L. L., & Han, T. T. (2012, May). Financial time series forecasting based on wavelet kernel support vector machine. In 2012 8th International Conference on Natural Computation (pp. 79-83). IEEE.

11.Dataset Source https://www.amarstock.com/csv-data-download12.

Dhaka_stock_market_prediction https://www.kaggle.com/code/heartbeat007/dhaka-stock-market-prediction