Dhaka Stock Exchange Stock Price Prediction using Machine Learning and Deep Learning Models

by

Debalina Barua
20266011

A thesis submitted to the Department of Computer Science and Engineering
in partial fulfillment of the requirements for the degree of
M.Engg. in Computer Science and Engineering

Department of Computer Science and Engineering
Brac University
September 2022

Declaration

It is hereby declared that

1. The thesis submitted is my/our own original work while completing degree at Brac University.

2. The thesis does not contain material previously published or written by a third party, except where this is appropriately cited through full and accurate referencing.

3. The thesis does not contain material which has been accepted, or submitted, for any other degree or diploma at a university or other institution.

4. We have acknowledged all main sources of help.

Student's Full Name & Signature:

Debalina Barua
20266011

Approval

The thesis/project titled "Dhaka Stock Exchange Stock Price Prediction using Machine Learning and Deep Learning Models" submitted by

1. Debalina Barua (20266011)

Of Summer, 2022 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of M.Engg. in Computer Science and Engineering on September 20, 2022.

Examining Committee:

Supervisor:
(Member)

Dr. Md. Golam Rabiul Alam, PhD
Professor
Department of Computer Science and Engineering
BRAC University

Program Coordinator:
(Member)

Dr. Amitabha Chakrabarty, PhD
Associate Professor
Department of Computer Science and Engineering
BRAC University

Head of Department:
(Chair)

Sadia Hamid Kazi, PhD
Chairperson and Associate Professor
Department of Computer Science and Engineering
BRAC University

Abstract

The stock market is unstable and generally unpredictable, as any one of us could have predicted. Researchers have been experimenting with time-series data to forecast future values for many years, with stock valuation forecasting being the most difficult and lucrative application. Market movement, however, depends on a variety of factors, only a small subset of which can be quantified, including historical stock data, trade volume, and current pricing. This makes predicting stock prices using machine learning difficult and, to some extent, unreliable. With an adequate amount of historical data and variables, mathematical and machine learning algorithms are used to anticipate short-term market movements for a typical, uninteresting market day. This paper proposes several comparative models for stock price prediction using various machine learning algorithms like Bidirectional LSTM, Multi-Head Attention-Based LSTM, Prophet, ARIMA etc. The models have been trained using historical data collected from the Dhaka Stock Exchange (DSE) official website. The financial data contains factors like Date, Volume, Open, High, Low Close, and Adj Close prices. The models are evaluated using standard strategic indicators like Mean Squared error (MSE), Mean Absolute Percentage Error (MAPE), Mean Absolute Error (MAE) and R-Squared. Moreover, in order to thoroughly understand the predictions, we implemented explainable AI models such as LIME. We believe that the information in this article will be useful to stock investors in determining the best times to buy and/or sell stocks on the Dhaka Stock Exchange.

Keywords: Bidirectional LSTM, Prophet, ARIMA, LIME,

Dedication

This work is dedicated to my beloved parents, loving husband and Dr. Md. Golam Rabiul Alam sir for guiding and bearing with me during this period with care and patience.

Acknowledgement

Firstly, we humbly praise and grateful to the Almighty, who permits us to live and accomplish tasks including the research work that is being presented here.
Secondly, it is my great pleasure to express immense gratitude and profound respect to my supervisor Dr. Md. Golam Rabiul Alam, Professor, Department of Computer Science and Engineering, BRAC University, for his constructive suggestions, scholastic guidance, enormous encouragement and kind cooperation throughout the entire progress of this research work. It would have been really difficult to bring this work towards completion without his invaluable advice. I also would like to thank all the concerned teachers and staff of the department for their direct and indirect assistance at different events of this work.
Finally, gratitude to my family and friends for their encouragement and unconditional moral support.

Table of Contents

List of Figures

Nomenclature

The next list describes several symbols & abbreviation that will be later used within the body of the document

AR Auto-regressive

$ARIMA$ Autoregressive Integrated Moving Average

$Bi-LSTM$ Bi-directional Long Short-term Memory

$BRNN$ Bidirectional Recurrent Neural Networks

CNN Convolutional Neural Network

DSE Dhaka Stock Exchange

GA Genetic Algorithm

GRU Gated Recurrent Unit

$LIME$ Local Interpretable Model-Agnostic

$LSTM$ Long short-term memory

MA Moving Average

MLP Multilayer Perceptrons

NN Neural Network

$PACF$ Partial Auto correlation

RNN Recurrent Neural Network

RNN Recurrent Neural Networks

ROI Return on Investment

SVM Support Vector Machine

SVR Support Vector Regression

Chapter 1

Introduction

Generally, 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.
Currently, stock forecasting models can be divided into two categories: traditional linear models and deep learning models. However, because time series data has both linear and nonlinear components, the results of forecasting models used alone are typically not very accurate. To greatly increase the accuracy and stability of the forecasting findings, several professionals and academics mix various single models. In this thesis, we predicted the future closing prices of five companies namely BPML, ASIAINS, SAIHAMCOT, ARAMITCEM and BEXIMCO, by feeding our system with historical stock price data of these companies. This prediction of the stock's upcoming price, if successful, could result in relevant profit. The goal of the paper is to provide investors with models that can effectively handle data with the right parameter values. Due to their capacity to deliver acceptable findings with the aid of technical examination of the data set, the models LSTM, Bi-LSTM, ARIMA and Prophet are provided. The goal is to evaluate the two models and use the one that is best suited for a certain company's data set.

1.1 Motivation

Sound capital market is a crucial component of the economy. Since the capital market gives businesses long-term funding, rapid economic progress may be slowed down in the absence of a healthy and effective capital market. Due to a lax regulatory environment, Bangladesh's capital market is still quite speculative and lacks transparency. Bangladesh's financial industry has historically been dominated by banks, and the capital market has less regulations since investors are generally discouraged by the capital market's risky behavior. However, the capital market began to exhibit lively behavior in the middle of the twentieth century, which piqued people's interest in stock exchanges. Many people began to invest their money in the hot market because everyone was making money and the index was increasing fast. Bangladesh stock market, during its life of six decades, has experienced two brutal crashes– one in 1996 and another during 2010-11 [2]. The presence of financial market bubbles is largely accepted nowadays. Understanding these inefficiencies and foreseeing when price bubbles will pop, however, is a very challenging endeavor. Imagine being able to spot a bubble before it forms and forecast when the market will crash. You would be able to sell at the perfect time to avoid losses in addition to making a profit while prices are rising. The data points to the emergence and fall of stock market bubbles as predictable phenomena; they happen frequently, and some of them are more harmful to the economy as a whole than others. When compared to the severe consequences of stock market bubble collapses, Bangladesh's existing body of knowledge in the topic has been determined to be insufficient. If the direction of the market is predicted well, investors will be better guided and the financial rewards will be substantial. The challenge in today's bad news environment is to be proactive rather than reactive. Therefore, construction companies try to predict stock prices, which must be taken into account in financial transactions to prevent sharp price drops. Time-series forecasting is a research field that aims to solve various problems, mainly in the financial field. It is worth noting that planning and decision-aiding tools are typically used in this area to minimize investment risk. Machine learning (ML) can come into its own and play a key role in a variety of critical applications. Therefore, by the help of Machine Learning and Deep Learning models, we can eradicate the financial losses faced by investors investing in stocks of wrong organizations.

1.2 Research Problem

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.

1.3 Aims and Objectives

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 used python scripting language 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. We will implement a mix of machine learning algorithms to predict the future stock price of this company, starting with simple algorithms like random forest, and then move on to advanced techniques like ARIMA, LSTM, bidirectional LSTM and Prophet.

1.4 Organization of the report

The report is structured in the following manner - In section II, I shall try to explain the concepts associated with the subject matter in detail. In area III, a short walk through the preceding research will be provided. Then, in part IV, I shall briefly explain the data collection methods and considerations of our proposed stock predicting dataset. Afterwards, in part V, the framework architecture and design choices will be reviewed. Finally, I'll wrap up with the experimental results and some future scopes of this lookup in area VI and section VII respectively.

Chapter 2

Background Study

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.

There are two ways to own a stock and make a profit, through dividends and capital appreciation. A dividend is a cash distribution of a company's profits. Principal appreciation is an increase in the stock price itself. Dividends are important to investors for five main reasons: they significantly boost stock investment earnings; they add another metric for fundamental research; they lower overall portfolio risk; they provide tax benefits; and they support the preservation of capital's buying power. In contrast to any ups and downs that may occur in the company's stock price over the course of a year, dividends are continuous, yearly indicators of a company's growth and profitability. A company that steadily generates profits and consistently raises its dividend payments over time is one that is less likely to have its fundamental

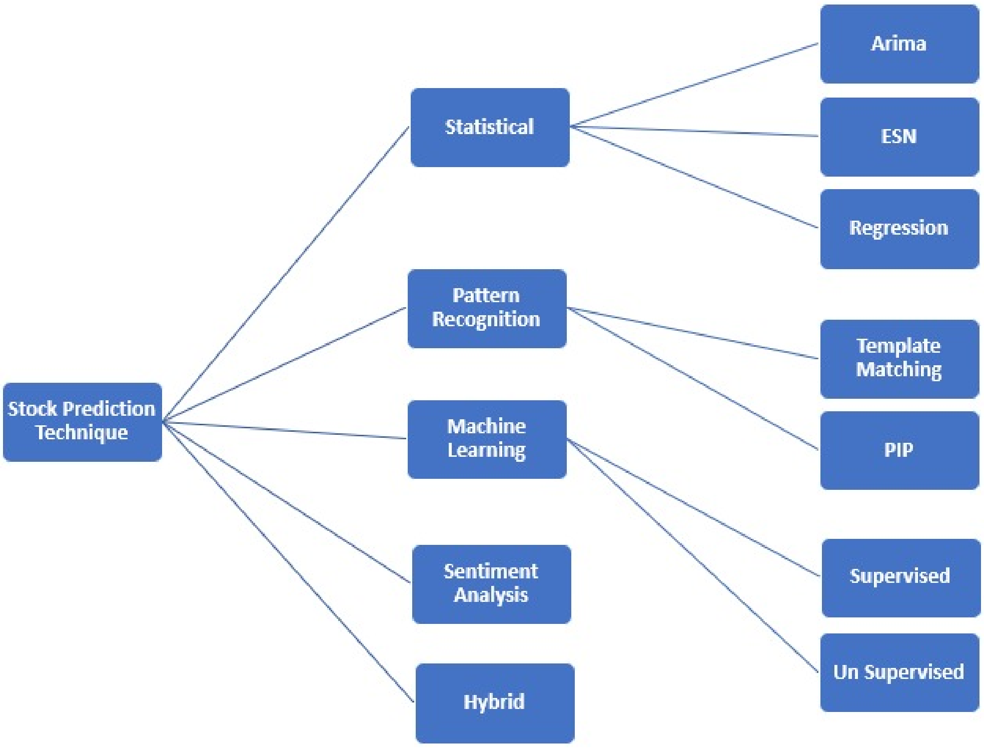

Figure 2.1: Taxonomy of Stock Prediction Techniques

financial stability endangered by transient market or economic downturns.

All ventures have a degree of risk. Stocks, bonds, shared stores, and exchange-traded stores can lose esteem in case showcase conditions decay. After you contribute, you make choices about what to do together with your budgetary resources. Your venture esteem might rise or drop since of advertised conditions or corporate choices, such as whether to grow into a modern region of commerce or blend with another company. Verifiably, stocks have outflanked most other speculations over the long run.

With predatory algorithms and other inside factors causing volatility and reversals that take advantage of the crowd's herd-like mentality, making money in the stock market is simpler than keeping it. It makes little sense to purchase stocks if they provide lower earnings than real estate or a money market account, which underscores the crucial issue of yearly returns. While historical evidence suggests that stocks can generate higher returns than other types of investments, long-term profitability necessitates risk management and strict discipline to stay clear of traps and occasional outliers.

The current portfolio theory offers an essential framework for managing wealth and assessing risk. whether you're a novice investor or have collected a substantial sum of money. This traditional market strategy is built on diversification, which warns long-term participants that depending solely on one asset class has a far higher risk than a basket filled with stocks, bonds, commodities, real estate, and other security kinds. We must also acknowledge that there are two different types of risk: systematic and unsystematic. The benefits of diversification are undermined by the high correlation between various asset types caused by the systemic risk from wars, recessions, and black swan events—unpredictable events with potentially disastrous effects.

The bottom line is stock markets are the beating heart of the market, and analysts frequently use stock prices as a gauge of the state of the economy. But stock markets are significant in more ways than only for speculating. Stock markets serve as a significant source of funding for publicly traded corporations by enabling businesses to sell their shares to hundreds or millions of ordinary investors.

Chapter 3

Related Work

There are many related researches on stock price prediction. The existing approaches are mainly divided into 2 categories: fundamental Analysis and Technical Analysis, to predict the longer term movement of a stock price. Interest rates, varied ratios of costs (such as price to earnings ratio), parameters relating to political economy form the premise for the fundamental Analysis whereas past price values, historical data, volumes listed over a specific time, various moving averages form the basis of Technical Analysis.

Then comes the machine learning approaches which incorporate varied machine learning techniques like Regression, ARIMA model, SVM, random-forest in addition to Neural Networks approaches. In order to predict various stock values at once, a multi-value associated network model of deep recurrent neural network based on LSTM (Associated Net) was proposed in [5]. By contrasting the model with the LSTM network model and the LSTM deep-recurrent neural network model, the viability and accuracy of the Associated Net were confirmed. To check whether the Associated Net model was applicable, several data sets were used. However, a popular method for stock market analysis is the ARIMA model [6]. ARMA blends moving average (MA) models that attempt to capture the shock effects seen in time series with auto-regressive (AR) models that attempt to explain the momentum and mean reversion effects frequently seen in trading markets. The ARMA model has a major flaw in that it ignores volatility clustering, a significant empirical phenomena in many financial time series. In [7], by merging basic and technical analytical variables from stock market indicators with the BP algorithm, a forward multi-layer neural network model for future stock price prediction was developed. The results demonstrate that this strategy outperforms the technical analysis method in terms of daily stock price prediction accuracy. An effective soft computing tech-nology was designed for Dhaka Stock Exchange (DSE) to predict the closing price of DSE [8]. [9] proposes an integrated framework in combination of wavelet transform and recurrent neural network based on artificial bee colony algorithm. With the use of Support Vector Machine, [10] proposed a stock market forecasting model. It is utilized in predicting and controlling decent subset features and overfitting for assessing stock indicators.

With the speedy progress of artificial intelligence, a lot of researchers use machine learning techniques for stock market forecasts. Support vector regression (SVR) may be a machine learning model that might determine the hyperplane during a high-dimensional space. In [10], SVR was accustomed to predict the gap value within the following day given the historical statistical data. Recurrent neural networks (RNNs) are a category of deep learning models for processing sequential data.

Third, the hybrid model research is as follows: Peter and Zhang used the hybrid ARIMA and ANN method to study the time series estimation [11]. In another paper, the authors proposed a hybrid machine learning system based on Genetic Algorithm (GA) and Support Vector Machines (SVM) for stock market prediction. However, the results showed that the hybrid GA-SVM system outperformed the stand alone SVM system [12]. In [13], they used a hybrid model that is a combination of two well-known networks, Long Short Term Memory (LSTM) and Gated Recurrent Unit (GRU). The prediction model provides less error by considering this random nature (change) for an outsized scale of data. In [14], auto-regressive integrated moving average (ARIMA), neural network (NN) and long short-term memory network (LSTM) were used to predict Bursa Malaysia's closing prices data. The accuracy of this model was more than 90%.

From DSE data, many research has been done. For instance, authors in [15], predicted stock prices of five selected companies from DSE using models such as SVM and ARIMA. Another paper [16], where researchers predicted stock prices from DSE using Support vector regression (SVR) and K-nearest neighbor (KNN) and compared both these models. The important motive of [17] was to predict the stock closing costs for two foremost stock exchanges in Bangladesh and evaluate the prediction accuracy primarily based totally on earlier than and after pandemic data. The applied models were Autoregressive Integrated Moving Average(ARIMA) and Support Vector Machine(SVM) and Long Short-Term Memory (LSTM). And it was inferred from the implementation that LSTM provided better accuracy than ARIMA and SVM for both the stock exchanges.In [18], the authors suggested a CNN-LSTM hybrid neural network with several parameters in this study to forecast stock values. Additionally, they incorporated an attention mechanism to raise the CNN-LSTM model's accuracy and scalability. In the trials, their suggested model was contrasted with several methods using data from two actual stock datasets. The outcomes support the effectiveness and scalability of our suggested approach.

To predict Chinese stock index, including the Shanghai composite index and the Shenzhen component index, the bagging method was used [19]. Each neural network was trained using the Adam optimization algorithm and the back propagation method, and the results demonstrate that the method has varying degrees of accuracy for predicting various stock indexes, but the prediction on close is unsatisfactory. In the realm of speech recognition, a multi-output speaker model based on RNN-LSTM was employed [20]. The results of the experiments demonstrate that the model is superior than one speaker models, and infrastructure fine-tuning when adding new output branches. Getting a new output model is preferable than training a new speaker model because it uses less memory.

The paper [21] proposes a CNN-BiLSTM-AM approach to forecast the closing price of stocks the next day. Convolutional neural networks (CNN), bi-directional long short-term memory (BiLSTM), and an attention mechanism make up this technique (AM). When compared to other methods, the CNN-BiLSTM-AM method is better suited for stock price prediction and for giving investors a trustworthy means to choose which stocks to buy. In order to estimate an organization's stock price using historical stock prices, this paper uses deep learning architectures such as Convolutional Neural Networks (CNN), Long Short-Term Memory (LSTM), Recurrent Neural Networks (RNN), Multilayer Perceptrons (MLP), and Support Vector Machines (SVM) [22]. It is clear from the comparing results that CNN outperforms the current models. Despite being trained on data from a single market, the network can predict stocks for multiple stock markets due to shared core dynamics between several stock markets. The comparison results show that the neural network model significantly outperforms the existing linear models when compared to them.

Chapter 4

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.

4.1 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)

4.2 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.

4.3 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!

Chapter 5

Methodology

5.1 Process Flow

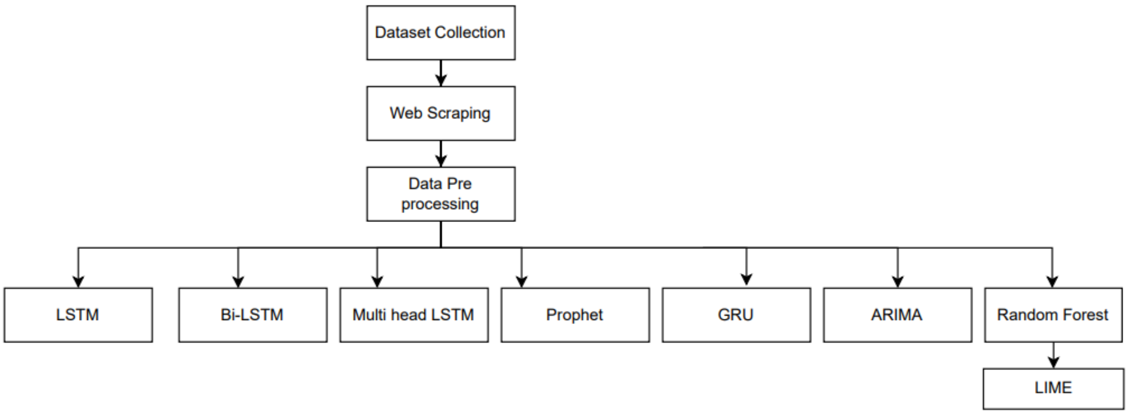

Figure 5.1: Work Flow Diagram

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

5.2 LSTM

Standard Recurrent Neural Networks (RNNs) are afflicted 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 timestep 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 timestep and mixing it with an entry of the present day one.

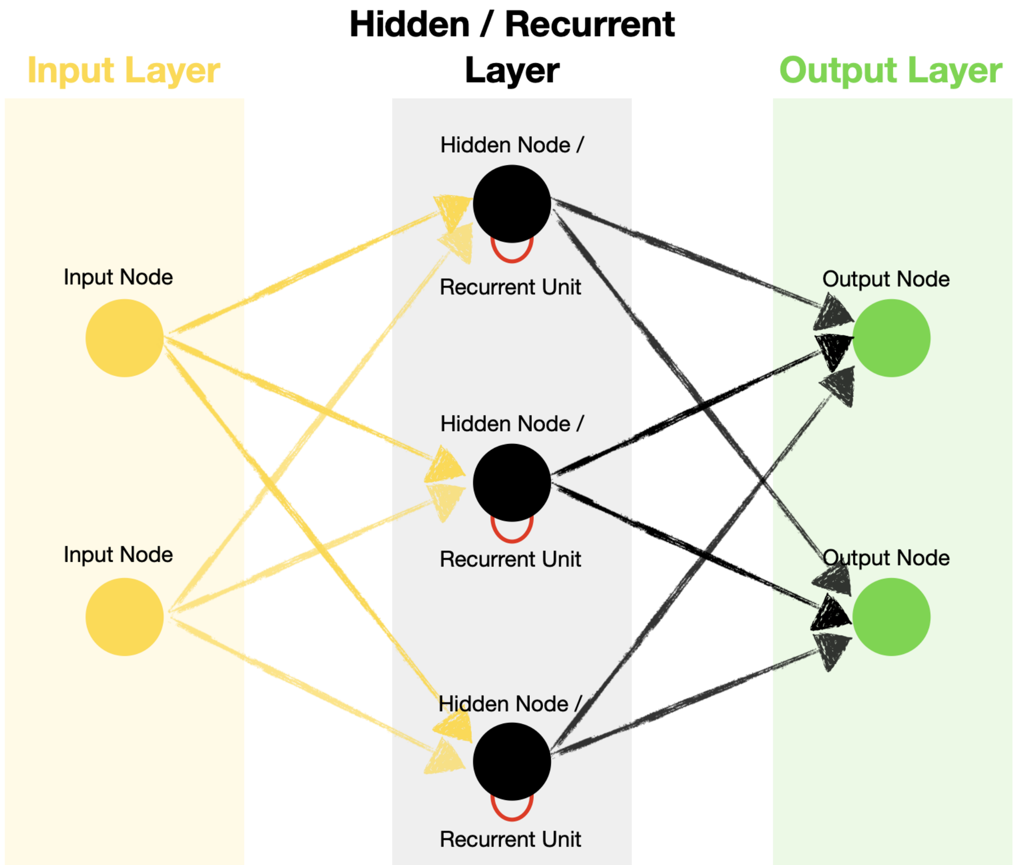

Figure 5.2: Standard Recurrent Neural Network architecture

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

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.

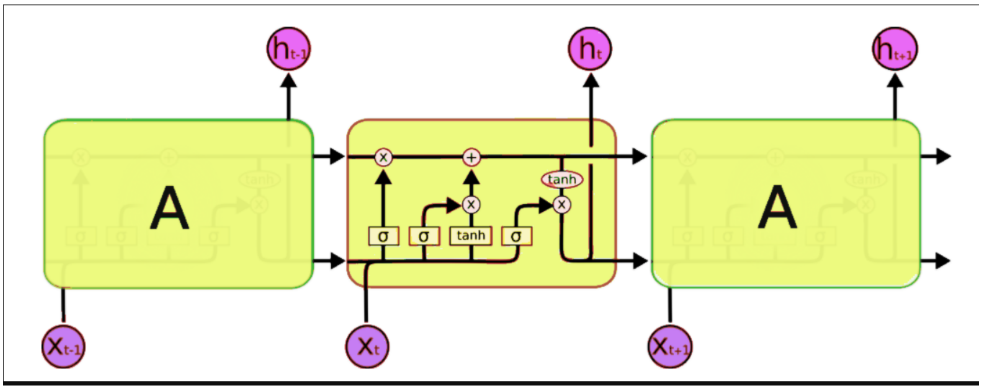

Figure 5.3: Architecture of LSTM

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.

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

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

Forget Gate: It finds 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.

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

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.

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

$$h_t = o_t * tanh(C_t) \tag{5.5}$$

5.3 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.

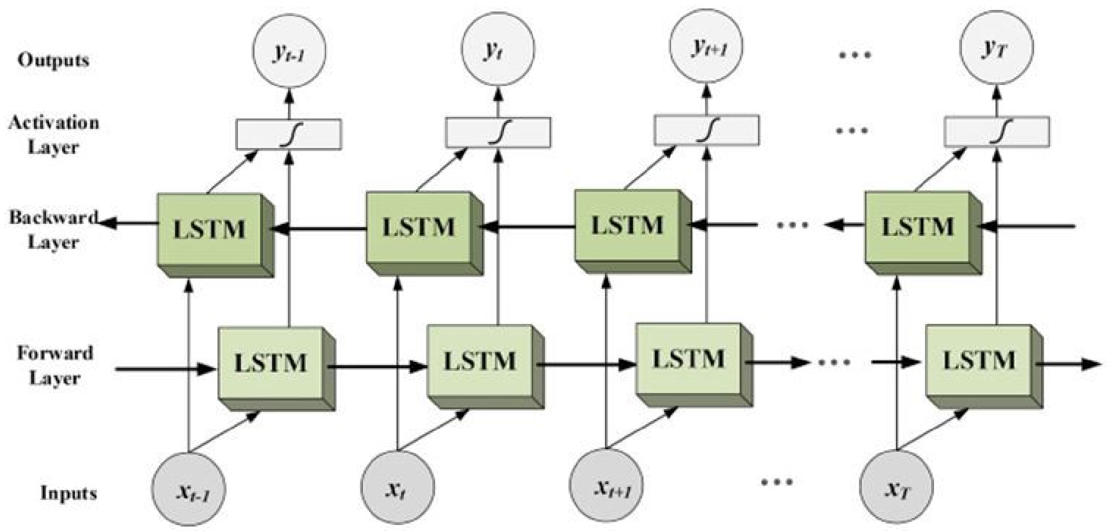

Figure 5.4: Bi-LSTM Neural Network Structure

5.4 Multiheaded Attention Based LSTM

The multi-head attention scheme was first 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:

$$s_i = softmax(Q_i \times K_i^H), s_i \in R^{B,1,T} \tag{5.6}$$

$$context_i = s_i \times V_i, context_i \in R^{B,1,\frac{Z}{n}} \tag{5.7}$$

$$CV = Concat([context_1, ..., context_n]), CV \in R^{B,1,Z} \tag{5.8}$$

where s_i represents the multi-head time-dimension attention score and $context_i$ represents the reduced-dimension context vectors from each subspace.

5.5 Gated Recurrent Unit (GRU)

GRUs and Long Short Term Memory are quite similar (LSTM). GRU uses gates to regulate the information flow, just like LSTM. When compared to LSTM, they are quite new. They have a simpler architecture and provide certain improvements over LSTM because of this. It is similar to an LSTM, but only has two gates - a reset gate and an update gate - and notably lacks an output gate as shown in figure 5.4. Fewer parameters means GRUs are generally easier/faster to train than their LSTM counterparts.

As with LSTMs, those gates are given sigmoid activations, forcing their values to lie in the interval (0,1). Intuitively, the reset gate controls how a whole lot of the preceding state we might still need to remember. Likewise, an replace gate could permit us to manipulate how a whole lot of the brand new state is only a copy of the old state
The Reset Gate is responsible for the short-term memory of the network i.e the hidden state (H_t). Here is the equation of the Reset gate.

$$r_t = \sigma(x_t * U_f + H_t - 1 * W_f) \tag{5.9}$$

The value of r_t will range from 0 to 1 because of the sigmoid function. Here U_r and W_r are weight matrices for the reset gate.

Similarly, we have an Update gate for long-term memory and the equation of the gate is shown below.

$$u_t = \sigma(x_t * U_u + H_t - 1 * W_u) \tag{5.10}$$

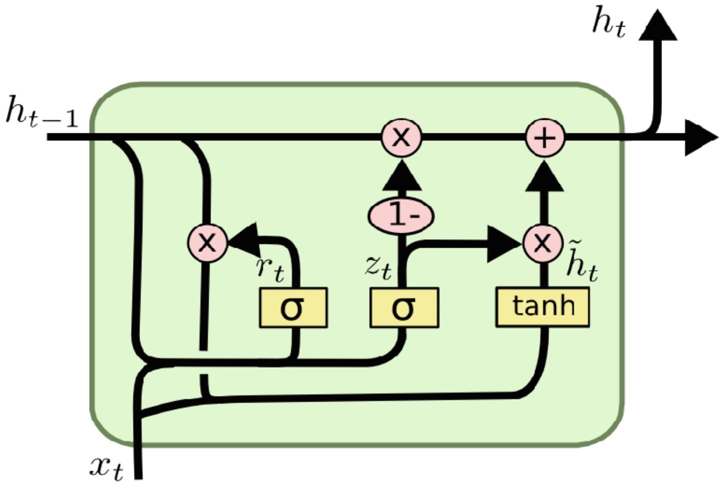

Figure 5.5: GRU Network

5.6 ARIMA

A model that uses time series data to make predictions is called an autoregressive integrated moving average (ARIMA). It is defined by the three terms p, d, and q, where p stands for the order of the AR term, q for the order of the MA term, and d for the order of differencing to convert a non-stationary series to stationary. Starting with autocorrelation (ACF) and partial autocorrelation (PACF) plot behavior, the process identifies models. After a model is defined, its parameters are estimated, and the Ljung-Box statistic test is used to do a diagnostic evaluation of the model's suitability. The model can start forecasting if it is sufficient.

The model calculation formula is as follows:

$$y_t = \theta_0 + \Phi_1 y_{t-2} + ... + \Phi_p y_{t-p} + \epsilon_t - \theta_1 \epsilon_{t-1} - \theta_2 \theta_{t-2} .... \theta_q \epsilon_{t-q},$$

$$\mathrm{y}_t = \theta_0 + \epsilon_1 y_{t-2} + ... + \epsilon_p y_{t-p} + \phi_t - \Theta_1 \epsilon_{t-1} - \theta_2 \epsilon_{t-2} -, , , , -\theta_q \epsilon_{t-q}, \quad (5.11)$$

where y_t and ϵ_t are the actual value and random error of the time period t, respectively; $\epsilon_i (i = 1, 2, \ldots, p)$ and $\theta_j (j = 1, 2, \ldots, q)$ are the model parameters; p and q, the order of the model (p and q are integers), are also the model parameter mentioned earlier; the random error ϵ_t, whose mean value is 0, is assumed to be independent and obey the same distribution in the model. The variance of constant term is denoted as σ_2 [25].

5.7 Prophet

Prophet is a technique for forecasting time series data from add-on models whose nonlinear patterns correspond to seasonal, weekly, daily, and holiday outcomes. It functions best when there are a few seasons of historical data and a number of time periods with strong seasonal results. The prophet frequently deals with outsiders and is strong on lost data and trend shifts. Compared to previous time series forecasting techniques, the prophet makes it clear how to generate a faster forecast that is more accurate. Compared to other models, this one requires very little computation time. To obtain forecasts in a matter of seconds, the prophet is comparable to the models in Stan. It enables us to obtain precise weather predictions from imperfect data without any manual effort. The Prophet has numerous “human” seasons of the week and time of year [26].

5.8 Random Forest

Popular machine learning algorithm Random Forest is a part of the supervised learning methodology. It can be applied to ML issues involving both classification and regression. It is built on the idea of ensemble learning, which is a method of integrating various classifiers to address difficult issues and enhance model performance.

The hyperparameters of a random forest are quite similar to those of a decision tree or a bagging classifier. Fortunately, using the classifier-class of random forest eliminates the requirement to combine a decision tree with a bagging classifier. Using the algorithm’s regressor, one may use random forest to handle regression problems as well.

The random forest algorithm works by four steps. First, select random samples from a given dataset and construct a decision tree for each sample and get a prediction result from each decision tree. Then a vote is performed for each predicted result. Finally, then the prediction result with the most votes as the final prediction is selected.

5.9 LIME

LIME, or Local Interpretable Model-Agnostic Explanations, is an algorithm that approximates any classifier or regressor locally with an interpretable model in order to explain the predictions of either in a faithful manner. It adjusts the feature values for a single data sample and then tracks how the changes affect the outcome. It acts as a ”explainer” to explain the conclusions drawn from each data sample. LIME produces a collection of explanations that show how each characteristic contributed to a prediction for a particular sample, which is an example of local interpretability.

Interpretable models in LIME include decision trees and linear regression, which are trained on minor alterations (such as noise addition, word deletion, or image concealment) to the original data. Using sample data points that are similar to the observation being explained, LIME tries to fit a local model. The local model may belong to the category of interpretable models, which includes decision trees, linear models, etc.

$$\xi_(x) = argmin_{g \in G} \zeta(f, g, \pi_x) + \Omega_(g) \tag{5.12}$$

Equation variables
f: an original predictor
x: original features
g: explanation model which could be a linear model, decision tree, or falling rule lists
π: proximity measure between an instance of z to x to define locality around x. It weighs z' (perturbed instances) depending upon their distance from x.
First Term: the measure of the unfaithfulness of g in approximating f in the locality defined by π. This is termed as locality-aware loss in the original paper
Last term: a measure of model complexity of explanation g. For example, if your explanation model is a decision tree it can be the depth of the tree or in the case of linear explanation models it can be the number of non zero weights.

Chapter 6

Implementation & 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 6.1:

6.1 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 6.1: 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 enters 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.

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 defined 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 statsmodels 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 autoregression, 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 fit 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.

6.2 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.

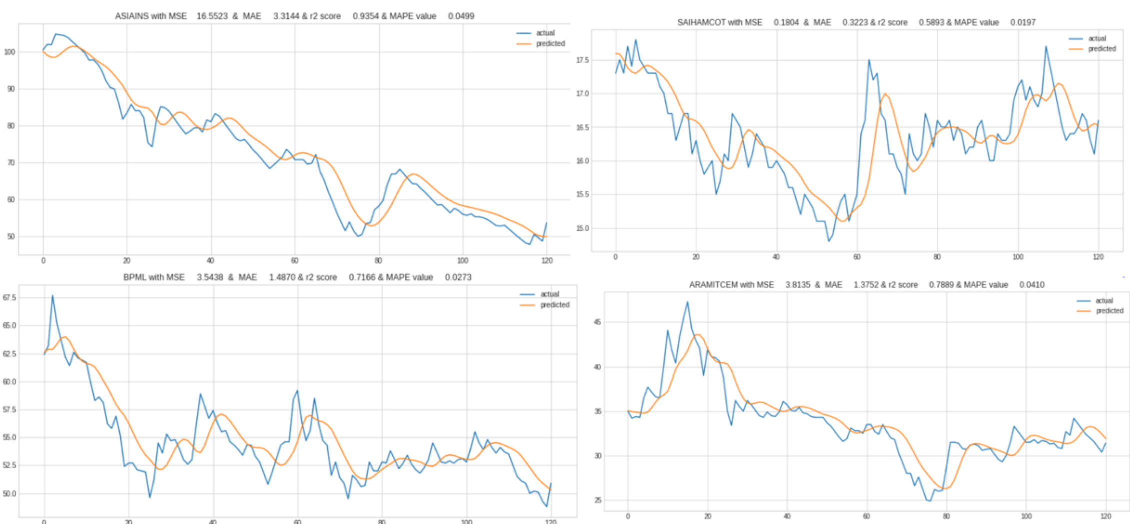

Figure 6.2: Results from LSTM Model

Figure 6.3: Results from LSTM Model on BEXIMCO

As shown in the graphs in Figure 6.2 and 6.3, we know that the lesser the MSE value, the smaller is the error and the better the estimator. Here, SAIHAMCOT which is reflected in Figure 6.3, has the least value when it comes to evaluating all the error functions. The closer to 0 the values are, the more accurate their predictions are. If we notice the graphs clearly, for a few company closing prices that we have predicted using LSTM, there are few peaks of actual price whereas the predicted price showed the opposite, smaller value.
However, we tried improving our prediction using lagging. Lags are very beneficial in time series analysis due to a phenomenon referred to as autocorrelation, that is a tendency for the values inside a time series to be correlated with preceding copies of itself. Now that we have checked the lag of our LSTM model that is shown in Figure 6.4, it is seen that the MSE value before lag was 3.54 whereas after lag it has decreased to 1.77.

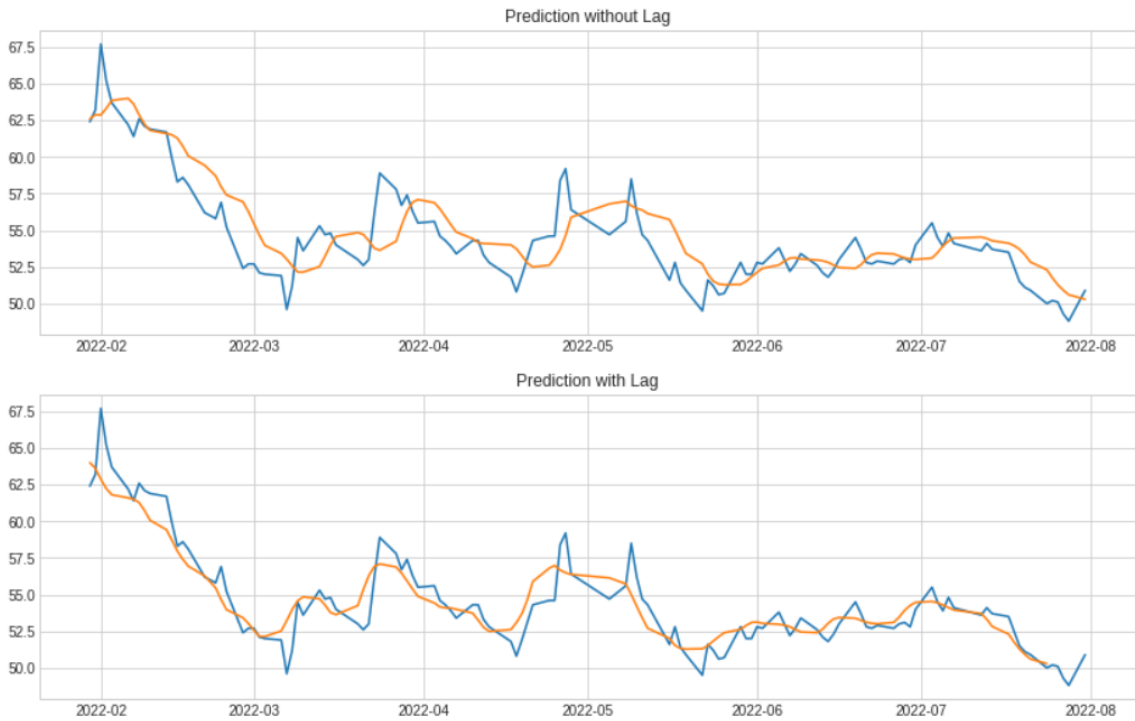

Figure 6.4: Checking lag of LSTM

If we look at the graph with lag in Figure 6.4, we can see that both the actual and predicted line graphs are nearly aligned with each other, meaning that our actual values and predicted values are almost the same.

Now, let us look at the predictions by our GRU model in Figure 6.5. If we compare GRU prediction results with LSTM model, we can see that the MSE, MAE, MAPE and r2 value for five of the companies increased when using GRU. Therefore, GRU will not be a suitable model for predicting when compared to LSTM.

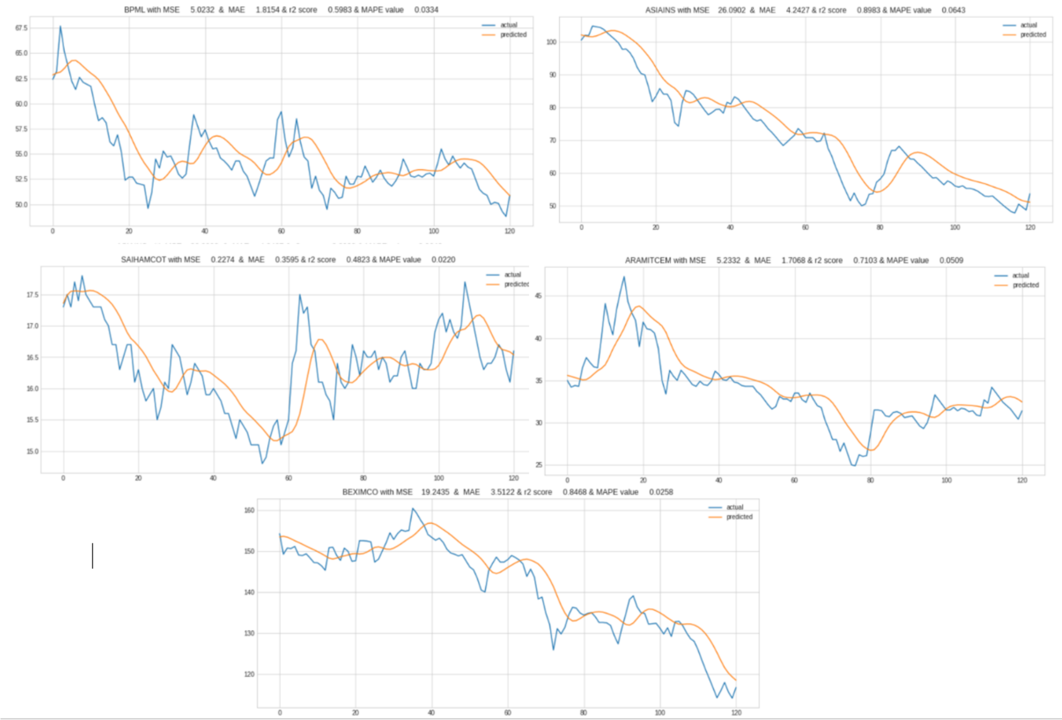

Figure 6.5: Results from GRU Model

Now looking at the predictions made using Bi-LSTM model in Figure 6.6, it is clear that if we follow the actual and predicted line graph, there is a huge difference between the two values. Comparing the MAPE values for the five companies with that of LSTM model MAPE values, we can see the values increased dramatically.

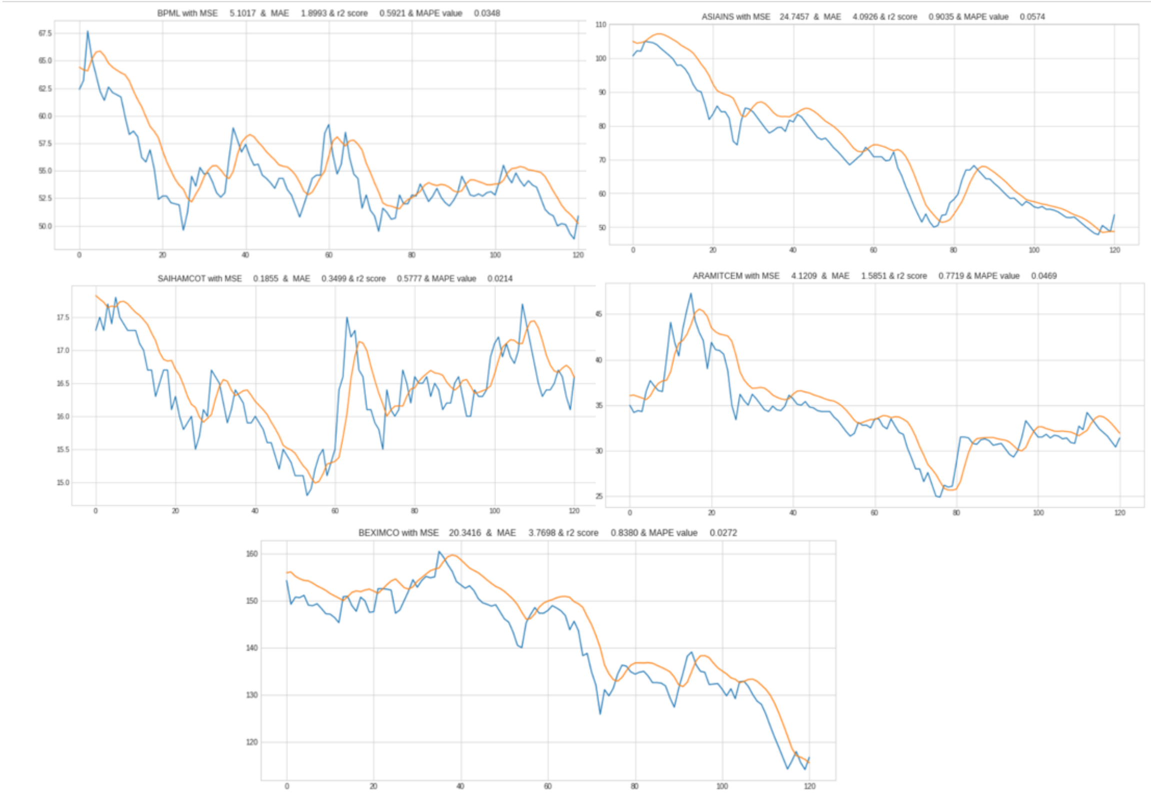

Figure 6.6: Results from Bi-LSTM Model

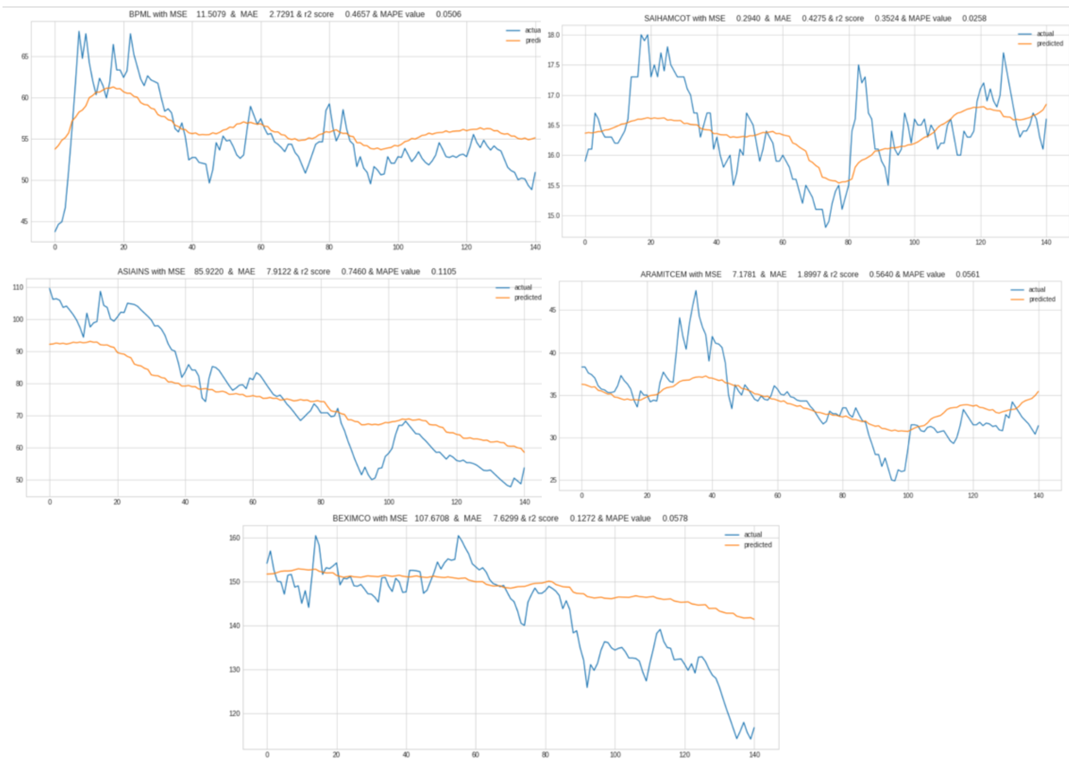

Figure 6.7: Results from Prophet Model

The prophet model under performed compared to Bi-LSTM model. Huge gap is observed between predicted and actual stock closing prices. It is clearly visible in the graphs shown in Figure 6.6.

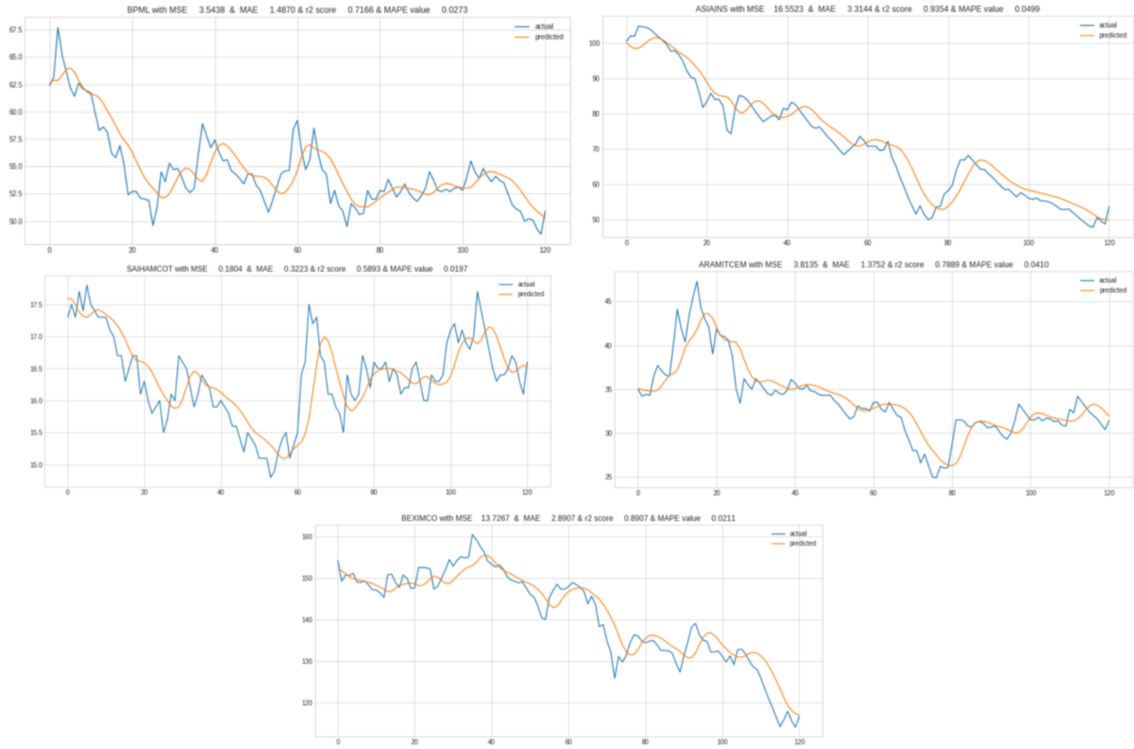

Figure 6.8: Results from ARIMA Model

Now we will compare our results from the LSTM model with the ARIMA Model. The graph speaks for itself if we look at the Figure 6.7. The predicted price and the actual price line graphs are as close as possible to each other. As a result, the MAPE values of ARIMA are lower than that of LSTM.

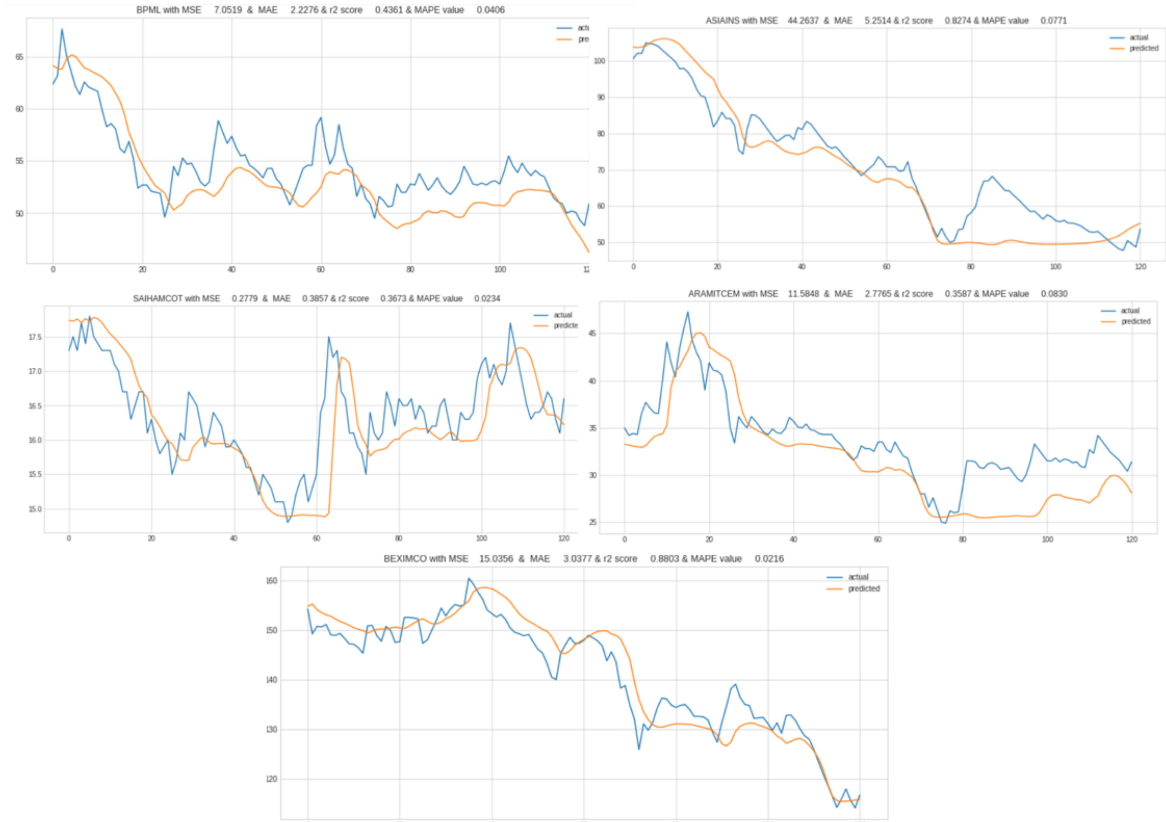

Figure 6.9: Results from Multihead-LSTM Model

Figure 6.10: Results from Lime of BPML and ASIAINS

Lastly, the last model to compare with is the Multihead LSTM as shown in Figure 6.9. And likewise, the error measures were higher than that of LSTM and ARIMA.

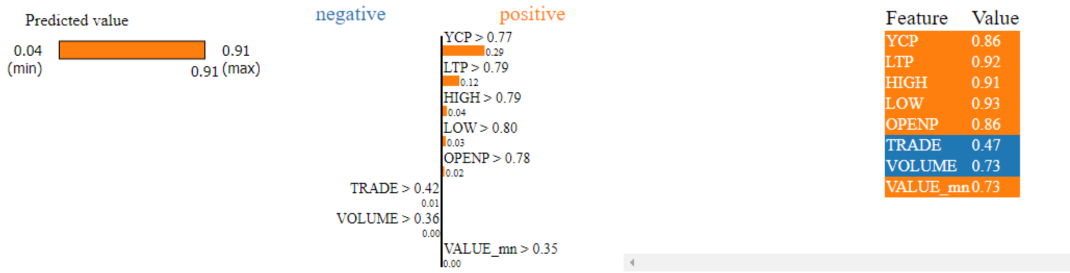

Figure 6.11: Results from Lime of BEXIMCO

The components of an interpretable model, such as the coefficients in a linear regression, that resemble the black-box model close to the point of interest and that have been trained over a new data representation to ensure interpretability are used by LIME to generate an explanation for a prediction. Now that we want to look at the explanation for stock ASIAINS with LIME Model interpreter based on Random Forest Regressor for target label Close Price as shown in Figure 6.10, it is observed that the predicted value of the closing price is 0.96 and the variables LTP, YCP, HIGH and LOW have positive influence while PENP, TRADE, VOLUME and VALUEmn have negative influence on predicted closing stock prices.

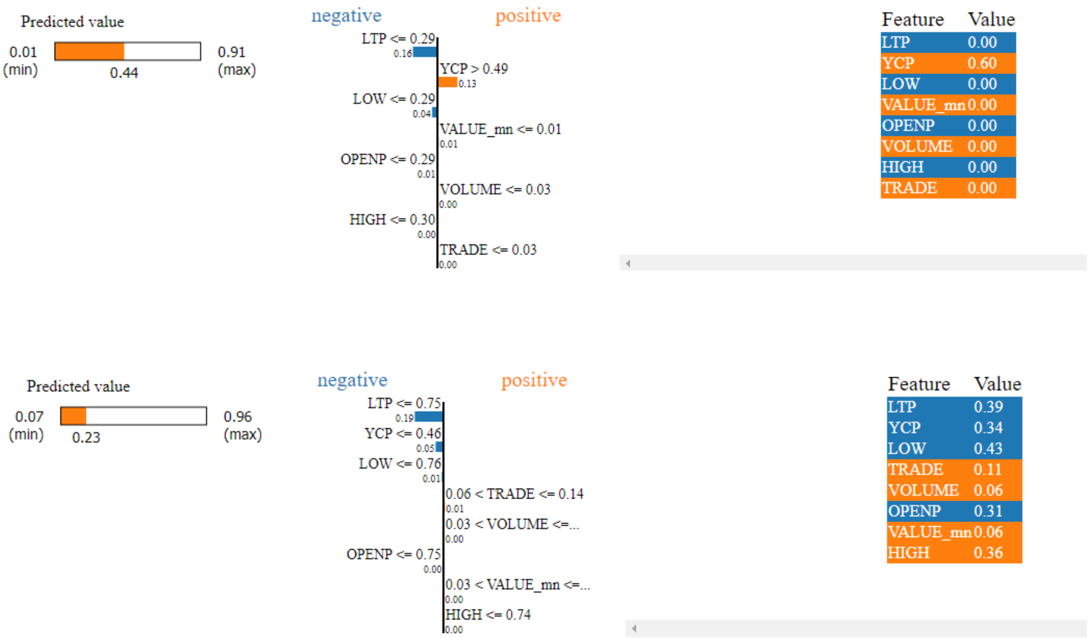

Figure 6.12: Results from Lime of SAIHAMCOT and ARAMITCEM

However, if we see the explanation for stock BEXIMCO in Figure 6.11, with LIME Model Interpreter, the variables TRADE and VOLUME have negative influence while YCP, LTP, HIGH, LOW, OPENP AND VALUEmn have negative influence on predicted closing stock prices. Moreover, in SAIHAMCOT, ASIAINS and ARAMITCEM mostly, as shown in Figure 6.12, four variables had positive influence and the other four had negative influences.

Chapter 7

Conclusion and Future Work

This paper establishes a forecasting framework to predict the prices of stocks. 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 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.

Our 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 denoised data. Moreover, in future more fine tuning of the models can provide us more accurate results in the future.

Bibliography

[1] S. Das, D. Mishra, and M. Rout, "A survey on impact of bio-inspired computation on stock market prediction," *Journal of Engineering Science and Technology Review*, vol. 10, pp. 104–114, Jul. 2017. DOI: 10.25103/jestr.103.15.

[2] M. Rahman and S. Hossain, "Stock market crash in bangladesh: The moneymaking psychology of domestic investors," *American Journal of Theoretical and Applied Business*, vol. 3, pp. 43–53, Sep. 2017. DOI: 10.11648/j.ajtab.20170303.12.

[3] F. S. Alzazah and X. Cheng, "Recent advances in stock market prediction using text mining: A survey," in *E-Business*, R. M. Wu and M. Mircea, Eds., Rijeka: IntechOpen, 2020, ch. 7. DOI: 10.5772/intechopen.92253. [Online]. Available: https://doi.org/10.5772/intechopen.92253.

[4] M. Vijh, D. Chandola, V. A. Tikkiwal, and A. Kumar, "Stock closing price prediction using machine learning techniques," *Procedia Computer Science*, vol. 167, pp. 599–606, 2020.

[5] G. Ding and L. Qin, "Study on the prediction of stock price based on the associated network model of lstm," *International Journal of Machine Learning and Cybernetics*, vol. 11, Jun. 2020. DOI: 10.1007/s13042-019-01041-1.

[6] H. M, G. E.A., V. K. Menon, and S. K.P., "Nse stock market prediction using deep-learning models," *Procedia Computer Science*, vol. 132, pp. 1351–1362, 2018, International Conference on Computational Intelligence and Data Science, ISSN: 1877-0509. DOI: https://doi.org/10.1016/j.procs.2018.05.050. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1877050918307828.

[7] A. A. Adebiyi, C. K. Ayo, M. O. Adebiyi, and S. O. Otokiti, "Stock price prediction using neural network with hybridized market indicators," 2012.

[8] T.-J. Hsieh, H.-F. Hsiao, and W.-C. Yeh, "Forecasting stock markets using wavelet transforms and recurrent neural networks: An integrated system based on artificial bee colony algorithm," *Applied Soft Computing*, vol. 11, pp. 2510–2525, Mar. 2011. DOI: 10.1016/j.asoc.2010.09.007.

[9] Y. Lin, H. Guo, and J. Hu, "An svm-based approach for stock market trend prediction," *The 2013 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–7, 2013.

[10] Y. Xia, Y. Liu, and Z. Chen, "Support vector regression for prediction of stock trend," *2013 6th International Conference on Information Management, Innovation Management and Industrial Engineering*, vol. 2, pp. 123–126, 2013.

[11] G. Zhang, “Time series forecasting using a hybrid arima and neural network model,” *Neurocomputing*, vol. 50, pp. 159–175, 2003, ISSN: 0925-2312. DOI: https://doi.org/10.1016/S0925-2312(01)00702-0. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0925231201007020.

[12] R. Choudhry and K. Garg, “A hybrid machine learning system for stock market forecasting,” *World Academy of Science, Engineering and Technology*, vol. 39, Jan. 2008.

[13] M. A. Hossain, R. Karim, R. Thulasiram, N. D. B. Bruce, and Y. Wang, “Hybrid deep learning model for stock price prediction,” in *2018 IEEE Symposium Series on Computational Intelligence (SSCI)*, 2018, pp. 1837–1844. DOI: 10.1109/SSCI.2018.8628641.

[14] M. K. Ho, H. Darman, and S. Musa, “Stock Price Prediction Using ARIMA, Neural Network and LSTM Models,” in *Journal of Physics Conference Series*, ser. Journal of Physics Conference Series, vol. 1988, Jul. 2021, 012041, p. 012 041. DOI: 10.1088/1742-6596/1988/1/012041.

[15] M. Hossain, S. Islam, P. Chakraborty, and A. Majumder, “Predicting daily closing prices of selected shares of dhaka stock exchange (dse) using support vector machines,” *Internet of Things and Cloud Computing*, vol. 8, pp. 46–51, Dec. 2020. DOI: 10.11648/j.iotcc.20200804.12.

[16] S. Islam, M. S. Sikder, M. F. Hossain, and P. Chakraborty, “Predicting the daily closing price of selected shares on the Dhaka Stock Exchange using machine learning techniques,” *SN Business & Economics*, vol. 1, no. 4, pp. 1–16, Apr. 2021. DOI: 10.1007/s43546-021-00065-. [Online]. Available: https://ideas.repec.org/a/spr/snbeco/v1y2021i4d10.1007_s43546-021-00065-6.html.

[17] T. F. Sinha, S. G. Rafia, M. A. Rahman, R. m. Rahat, R. H. Nabil, and A. Bhowmik, “Stock market comparison and analysis in preceding and following pandemic in bangladesh using machine learning approaches,” in *Proceedings of the 2nd International Conference on Computing Advancements*, ser. ICCA ’22, Dhaka, Bangladesh: Association for Computing Machinery, 2022, pp. 260–268, ISBN: 9781450397346. DOI: 10.1145/3542954.3542993. [Online]. Available: https://doi.org/10.1145/3542954.3542993.

[18] C. Li, X. Zhang, M. Qaosar, S. Ahmed, K. M. R. Alam, and Y. Morimoto, “Multi-factor based stock price prediction using hybrid neural networks with attention mechanism,” in *2019 IEEE Intl Conf on Dependable, Autonomic and Secure Computing, Intl Conf on Pervasive Intelligence and Computing, Intl Conf on Cloud and Big Data Computing, Intl Conf on Cyber Science and Technology Congress (DASC/PiCom/CBDCom/CyberSciTech)*, 2019, pp. 961–966. DOI: 10.1109/DASC/PiCom/CBDCom/CyberSciTech.2019.00176.

[19] B. Yang, Z.-J. Gong, and W. Yang, “Stock market index prediction using deep neural network ensemble,” *2017 36th Chinese Control Conference (CCC)*, pp. 3882–3887, 2017.

[20] S. Pascual and A. Bonafonte, “Multi-output rnn-lstm for multiple speaker speech synthesis and adaptation,” *2016 24th European Signal Processing Conference (EUSIPCO)*, pp. 2325–2329, 2016.

[21] W. Lu, J. Li, J. Wang, and L. Qin, “A cnn-bilstm-am method for stock price prediction,” *Neural Comput. Appl.*, vol. 33, pp. 4741–4753, 2021.

[22] C. N. Anand, “Comparison of stock price prediction models using pre-trained neural networks,” *March 2021*, 2021.

[23] D. Xiao and J. Su, “Research on stock price time series prediction based on deep learning and autoregressive integrated moving average,” en, *Sci. Program.*, vol. 2022, pp. 1–12, Mar. 2022.

[24] A. Vaswani, N. Shazeer, N. Parmar, *et al.*, “Attention is all you need,” in *Advances in Neural Information Processing Systems*, I. Guyon, U. V. Luxburg, S. Bengio, *et al.*, Eds., vol. 30, Curran Associates, Inc., 2017. [Online]. Available: https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

[25] G. Van Houdt, C. Mosquera, and G. Nápoles, “A review on the long short-term memory model,” *Artificial Intelligence Review*, vol. 53, Dec. 2020. DOI: 10.1007/s10462-020-09838-1.

[26] Kaninde, Sumedh, Mahajan, Manish, Janghale, Aditya, and Joshi, Bharti, “Stock price prediction using facebook prophet,” *ITM Web Conf.*, vol. 44, p. 03 060, 2022. DOI: 10.1051/itmconf/20224403060. [Online]. Available: https://doi.org/10.1051/itmconf/20224403060.