CHAPTER 1

INTRODUCTION

Stock market is characterized as dynamic, unpredictable and non-linear in nature. Predicting stock prices is a challenging task as it depends on various factors including but not limited to political conditions, global economy, company's financial reports and performance etc. Thus, to maximize the profit and minimize the losses, techniques to predict values of the stock in advance by analyzing the trend over the last few years, could prove to be highly useful for making stock market movements. Two methods that are widely used in general are namely Fundamental Analysis and Technical Analysis.

Fundamental Analysis: To determine accurate product value, reliable and accurate information on the financial report of the company, it is necessary to have competitive strength and economic conditions in which they are interested. The above value of the product can be used to make an investment decision. On the basis of this idea, "if the intrinsic value is higher than the market value it holds, invest otherwise and avoid it as a bad investment". Not only are these parameters other parameters such as book value, earnings, p/e ratio, ROI etc. should be carefully analyzed to obtain an estimate of future business conditions. For the long-term predictions, Fundamental analysis is useful and the advantages are due to their systematic approach and their ability to predict changes.

Technical Analysis: "The idea behind technical analysis is that investors' constantly changing attributes in response to different forces/factors make stock prices trends/movements". Different technical factors of quantitative parameters can be used for analysis, such as trend indicators, lowest and highest daily values, indices, daily ups and downs, stock volume, etc. It is possible to extract rules from the data and the investors make future decisions based on these rules.

Different analysts may derive from the same charts different rules. For both short and long term analysis, technical analysis is used. Technical analysis data is preferable over fundamental analysis data as input to system. Now a days, advanced intelligent techniques based on either technical or fundamental analysis are used for predicting stock prices. Particularly, for stock market analysis, the data size is huge and non-linear. To deal with this variety of data efficient model is needed that can identify the hidden patterns and complex relations in this large data set. Machine learning techniques in this area have proved to improve efficiencies by 60-86 percent as compared to the past methods. Although, AI has given us the ability to analyze both technical and fundamental with the help of the ensemble.

For many business analysts and researchers, forecasting the stock market price is always a challenge. Stock market prices estimation is not only an interesting but also challenging area of research. Predicting the stock market with full accuracy is very difficult as external entities such as social, psychological, political and economic have a great and substantial influence on it. The main characteristic of the data associated with stock market is usually time variant and nonlinear. Prediction of stock market plays a vital role in stock business. If investors lack sufficient information and knowledge then their investment can suffer the greatest loss. Investors must predict the future stock value of companies in order to obtain high profits. Various prediction techniques have been developed to do predictions on the stock market accurately. There were two methods widely known as conventional methods at the time when there were no computational methods for risk analysis. There are many conventional methods for predicting stock prices (by analyzing past data).

Several Machine Learning and Deep Learning techniques have been used independently for stock price prediction. They seem to work well only for a short time span and the key feature of generalization (i.e. to perform well on previously unseen data) is lost. This is due to a martingale effect on the stock price. This has made such techniques well suited for short term gains. Recent improvements and ideas in the field of AI have given rise to better models with state-of-the-art performance and have the potential to get previously unseen results.Ensemble Techniques also called bagging and stacking is one of those ideas that has wandered around for a long-time. This technique is proven and is followed by data scientists since it gives better results than individual models. The intuition behind this is that individual models after ensemble improve each other mistakes since they might learn different features about the data while training. This also stands true for stock market prediction. Better performance has also given rise to higher computations requirements. Recent improvements in computation handling and efforts from the community have given us techniques which have made a benchmark in their respective fields. These techniques are ensembles from inter domains as well as intra- domains.

1.1 IMPORTANCE OF STOCK MARKET

Indian stock market stood at third rank in the world. The Stock is essentially a share in a company’s ownership. Stocks are partial ownership of businesses instead of stock tickers piece of paper, which can be traded in the stock market. If company ownership is divided into 100 parts, the investor purchase one part which is equal to one share then we can own 1 percent of that company. Stock exchange uses an automated matching system driven by order. Stock prices are defined as any time how many buyers and sellers available for the same stock in the market. If the number of buyers is more than sellers then stock price becomes high and if the number of sellers higher than buyers then stock price becomes low. The best buy and sell order are looked into a counterparty angle. The best buy order is which has the highest price and best sell order is which has the lowest price. With this logic system can match the orders and executes the traders' system. SEBI (Security and Exchange Board of India) regulates the stock market. In stock markets customers preferences and requirements are different. The estimated world stock market was at $36.6 trillion in early October 2008. The total world market for derivatives was estimated at approximately $791 trillion in face value or nominal value, 11 times the size of the world economy.

1.2 DEEP LEARNING

Deep learning is part of a broader family of machine learning methods, which is based on artificial neural networks with representation learning. Learning can be supervised, semi-supervised or unsupervised. Deep learning is a class of machine learning algorithms that uses multiple layers to progressively extract higher-level features from the raw input. For example, in image processing, lower layers may identify edges, while higher layers may identify the concepts relevant to a human such as digits or letters or faces.

From another angle to view deep learning, deep learning refers to 'computer-simulate' or 'automate' human learning processes from a source (e.g., an image of dogs) to a learned object (dogs). Therefore, a notion coined as "deeper" learning or "deepest" learning makes sense. The deepest learning refers to the fully automatic learning from a source to a final learned object. A deeper learning thus refers to a mixed learning process: a human learning process from a source to a learned semi-object, followed by a computer learning process from the human learned semi-object to a final learned object. In deep learning, each level learns to transform its input data into a slightly more abstract and composite representation. In an image recognition application, the raw input may be a matrix of pixels; the first representational layer may abstract the pixels and encode edges; the second layer may compose and encode arrangements of edges; the third layer may encode a nose and eyes; and the fourth layer may recognize that the image contains a face. Importantly, a deep learning process can learn which features to optimally place in which level on its own. This does not eliminate the need for handtuning; for example, varying numbers of layers and layer sizes can provide different degrees of abstraction.

Applications of Deep Learning Across Industries:

1. Self Driving Cars
2. News Aggregation and Fraud News Detection
3. Natural Language Processing
4. Virtual Assistants
5. Entertainment
6. Visual Recognition
7. Fraud Detection
8. Healthcare
9. Personalisations
10. Finance

CHAPTER 2

LITERATURE SURVEY

There were two important indicators in the literature for stock price forecasting. They are fundamental and technical analysis. Both were used to analyse the stock market.

2.1 TECHNICAL ANALYSIS

[A] Stock Forecasting Method Based On Wavelet Analysis And ARIMA-SVR Model.

Published in: 2021 3rd International Conference on Information Management (ICIM)

Tian Ye used wavelet analysis and ARIMA-SVR for stock market price prediction. The stock price was decomposed into reconstructed part and error part by wavelet decomposition and wavelet reconstruction. Then, the ARIMA model and the SVR model are used to forecast the reconstructed part and the error part respectively. The models were created and tested on the closing price of the Shanghai Pudong Development Bank from January 5, 2019, to January 29, 2021, with a 90%-10% train-test-split. The results were considered satisfactory with MSE of 0.57.

[B] Restricted Boltzmann Machines For The Prediction Of Trends In Financial Time Series.

Published in: 2021 International Joint Conference on Neural Networks (IJCNN)

Carlos A. S. Assis; Adriano C. M. Pereira; Eduardo G. Carrano; Rafael Ramos; Wanderson Dias proposed the use of Restricted Boltzmann machine for predicting the stock price. The approach comprised of five steps: extraction of historical data; transformation; dimensionality reduction and feature extraction; classification and analysis of result the approach was applied to five real time series from BM&F BOVESPA. The result was found for the combination of RBM and SVM being 54% - 66% accurate.

[C] Cross-Domain Deep Learning Approach For Multiple Financial Market Prediction.

Published in: 2020 International Joint Conference on Neural Networks (IJCNN).

Xinxin Jiang, Shirui Pan, Jing Jiang and Guodong Long applied cross domain deep learning approach for multiple financial market prediction. Attention mechanisms were applied on different models and used on currency domain and stock domain. The approach was tested on currency markets and stock markets of USA, China and India before and after financial crisis period. Performance evaluation was using F1 Score and Area under Curve (AUC).

[D] A Comparative Study On The Individual Stock Price Prediction With The Application Of Neural Network Models.

Published in: 2021 International Conference on Computer Engineering and Artificial Intelligence (ICCEAI)

Chen, Zhou & Dan (2021) use a LSTM model on Chinese Stock market data for making predictions. The training data has been sampled from time periods which provide varying amount of returns. In all the surveyed studies, LSTM models are used scarcely and very few studies pre-process the data thoroughly. This greatly alters the performance of the model. This study fills that gap. Standardisation on the data improves the performance of the models by centring noise from trend reversal signals and normalisation prevents the model weights from being skewed. This study introduces the use of GRU models and ICA for price predictions and optimises the performance of LSTM models. These techniques have not been used before.

2.2 FUNDAMENTAL ANALYSIS

[A] Using sentimental analysis in prediction of stock market investment

Published in: 2016 5th International Conference on Reliability, Infocom Technologies and Optimization (ICRITO)

The paper discussed SM moves that have a considerable influence on the market value of particular businesses due to domestic and global considerations. The research looked at three different features of Brazilian social media behaviour on Twitter: (A) total number of Tweet emotions; (B) Tweet sentiments with likes; and (C) Balanced sentiments should be acknowledged. In their endeavour to create SA in Portuguese, they employed directly the Multilayer Perceptron methodology.

[B] Research on the text sentiment classification about the social hot events on Weibo.

Published in: 2020 10th International Symposium on Chinese Spoken Language Processing

J. Lin, A. Yang, Y. Yong studied the sentiment classification about public comments of the social hot event on Weibo. First, author put forward the sentiment dictionary-based classification method and the accuracy rate is close to 50%. In order to improve the accuracy,they also proposed a sentiment classification method based on Naïve Bayesian and used TF as the feature weight and chi-square test to extract features of each class. The accuracy rate is raised to more than 70%. It's better than present method aiming to the whole unified text on Weibo and has remedied the shortage of sentiment analysis aiming to typical events on Weibo.

[C] Sentiment Analysis Of Stock Markets Using A Novel Dimensional Valence–Arousal Approach.

Published in: 2021 International Conference on Web Search and Data Mining.

Wu et al. proposed a deep learning (DL) method for the prediction of the stock dimensional valence-arousal sentiments in the stock market. The method used the title, keywords, and overview of stock market-related messages for estimation of all vectors using the hieratical attention approach. The method achieved success, producing better results. However, it cannot identify the words with multiple meanings, and it also needs some stability improvements. Similar to the aforementioned technique, a DL-based method was employed in for extrapolation of Stock market using sentiment analysis. The model is based on RNN and LSTM techniques which is then utilized to define the sentiments into positive and negative class. The increase or decrease in stock prices is predicted from sentiment analysis.

CHAPTER 3

PROBLEM STATEMENT AND SOLUTION STRATEGY

Forecasting stock market is one of most sophisticated job. Nowadays, uncertainty characterizes the financial and economic environs, and thereby, understanding the predictability of future asset returns is the trending research topic for professionals and academics. The forecasting of the stock price attempts to assess the potential movement of a financial exchange's stock value. The accurate price movement would contribute more benefit to investors in production. However, the most challenging aspect is the forecasting of stock movement as it involves the factors like interest rates, politics, economic growth, etc. The hybrid optimization algorithms have been reported to be promising for certain search problems. This makes the use of machine learning and deep learning models for stock forecasting with the training of price fluctuations of days and even the minute. Models like LSTM, GRU and ARIMA are more common in forecasting stock prices. To be more accurate, the Metaheuristic tactics are incorporated with the learning algorithm.

Investors are familiar with the saying, “buy low, sell high” but this does not provide enough context to make proper investment decisions. Before an investor invests in any stock, he needs to be aware how the stock market behaves. Investing in a good stock but at a bad time can have disastrous results, while investment in a mediocre stock at the right time can bear profits. Financial investors of today are facing this problem of trading as they do not properly understand as to which stocks to buy or which stocks to sell in order to get optimum profits. Predicting long term value of the stock is relatively easy than predicting on day-to-day basis as the stocks fluctuate rapidly every hour based on world events.

3.1 EXISTING SYSTEM

The other existing approaches makes use of Neural Networks. Neural networks have the following drawbacks:

3.1.1 Local Minima and Maxima :

Neural Networks are based on gradient descent method to find the local extreme value and they have a tendency to get stuck on the local minima and maxima and therefore it is difficult to find global minima and maxima.

3.1.2 Slow Convergence Rate :

The neural network takes a lot of time to train. The most existing systems focus only on technical analysis not on fundamental analysis but stock market forecasting is not only about analyzing the historical price of stock but also analyzing the investor mindset, company profile, market situation, economic and global factors.

3.2 SOLUTION STRATEGY

The Solution to resolve the above problem is using a different algorithm and different technique to perform the prediction. We will implement the system using three different modules. One using deep learning modules that includes LSTM(Long Short Term Memory) algorithm, ARIMA(Auto-Regressive Integrated Moving Average) algorithm and Linear Regression algorithm and second using sentimental analysis module for fundamental analysis of tweets and news that affect stock market and third implementation using Rainbow DQN agent module that gives recommendation on stock using tweet polarity and forecasted results that is obtained from both the previous modules. Finally ensembling them into single architecture to provide a sufficient accuracy to predict the stock market.

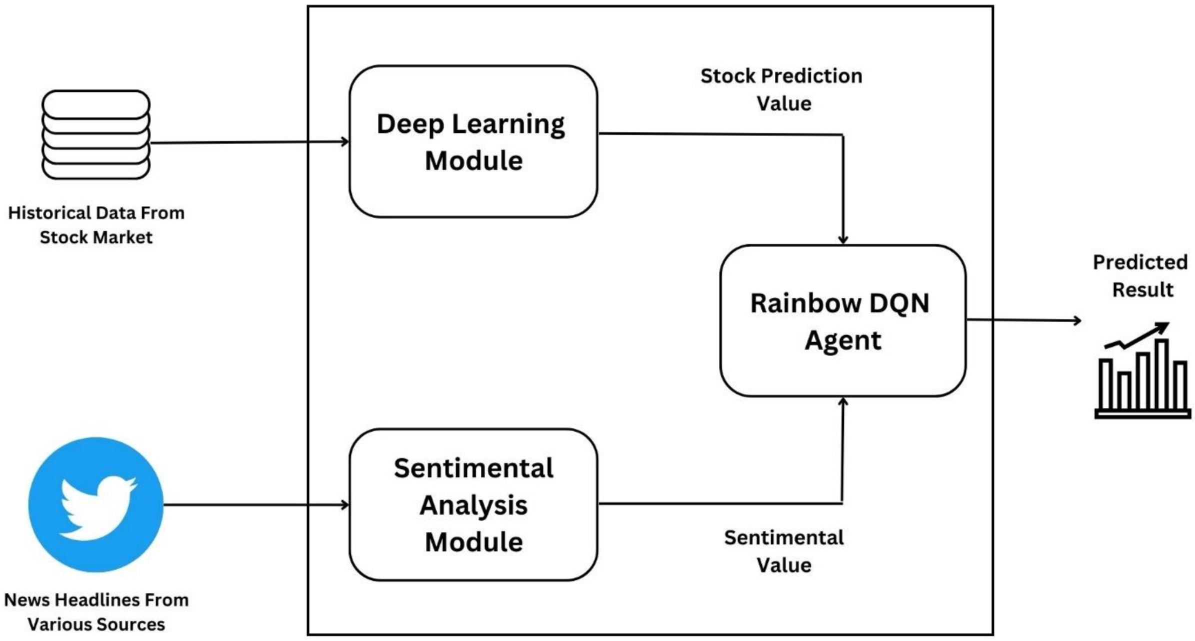

Fig 3.1 : Solution Strategy

CHAPTER 4

PROPOSED SYSTEM

The proposed system uses ensemble techniques which is a mix of both deep learning and machine learning techniques such as LSTM (Long Short Term Memory) algorithm ARIMA (Auto-Regressive Integrated Moving Average) and Linear Regression to predict the market prices. This system extends to make sentimental analysis on text of twitter and various news headlines which helps in fundamental analysis and also uses improved time interval of size 1min and 5min to update the data from various sites to predict the future movements. The App forecasts stock prices of the next seven days for any given stock under NASDAQ or NSE as input by the user. Predictions are made using three algorithms: ARIMA, LSTM, Linear Regression.

This Project combines the predicted prices of the next seven days with the sentiment analysis of tweets to give recommendation whether the price is going to rise or fall. we use the ensemble approach on deep learning algorithms considering technical and fundamental parameters for training 3 different models (i.e. Linear Regression, LSTM and ARIMA) to combine and achieve the features from both models for real-time stock market prediction. The models will perform well on short term price prediction as compared to long term price prediction with appreciable directional accuracy. We also proposed the use of a state-of-the-art technique Rainbow DQN for the prediction of buy/sell signals which outperforms all the existing systems in terms of Return of Investment and RMSE(Root Mean Square Error)

Advantages of this system :

- This system gives the best result based on accuracy.
- This system outperforms all the existing systems in terms of Return of Investment.
- This system reduces market risk and helps individual to gain profit.

4.1 METHODOLOGY

1) **Data Collection :** In this step raw data from stock market is fetched through Yahoo finance library and the news and tweets from twitter is also collected using alpha vantage and tweepy library.
2) **Data Preprocessing:** Here the stock data collected is processed using numpy and pandas and tweets collected is processed using tweet processor of anaconda3 and textblob library.

3) **Feature Selection:** It is a process used in machine learning and statistics to select a subset of relevant features (variables, predictors) for use in model construction. Here features like open, close, high, low etc.. are selected.
4) **Preprocessed Data:** In this step the raw data is cleaned and processed into xml format with dimensionality reduction and tweets are cleaned and tokenized using textblob and nltk package.
5) **DNN Model Selection:** Here the preprocessed data is fed into three DNN model such as ARIMA, LSTM, Linear Regression.
6) **Forecasted Result:** In this step the three Deep learning models forecasts the closing price of selected stock for next 7 days. Meanwhile the sentimental score from processed tweets is obtained.
7) **Rainbow DQN Agent:** The DQN Agent is an off-policy deep reinforcement learning algorithm which takes sentimental score and tweet polarity as input and gives the stock recommendation.
8) **Buy/Sell Signal:** Finally from the above agent the recommendation of stock is obtained which gives buy or sell signal of particular stock to the user.

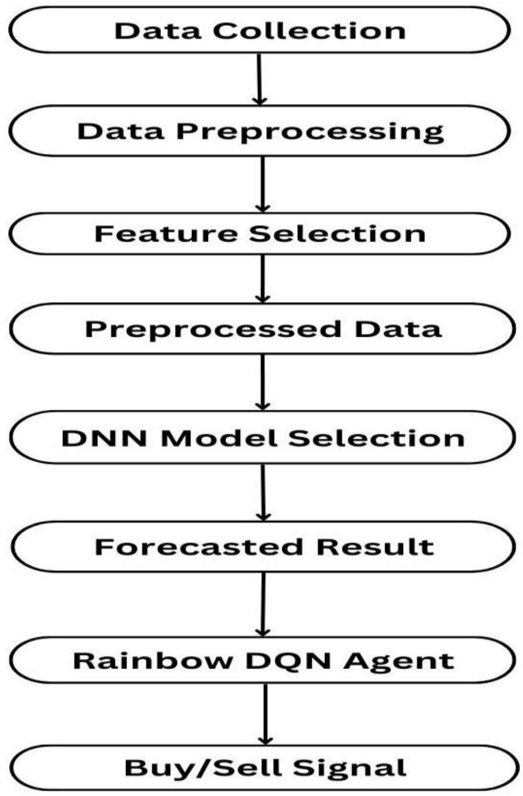

Fig 4.1 : Steps In Forecasting

4.2 MACHINE LEARNING

Machine learning (ML) is the study of computer algorithms that improve automatically through experience. It is seen as a subset of artificial intelligence. Machine learning algorithms build a mathematical model based on sample data, known as "training data", to make predictions or decisions without being explicitly programmed to do so. Machine learning algorithms are used in a wide variety of applications, such as email filtering and computer vision, where it is difficult or infeasible to develop conventional algorithms to perform the needed tasks. Early classifications for machine learning approaches sometimes divided them into three broad categories, depending on the nature of the "signal" or "feedback" available to the learning system.

These were:

1. Supervised learning: The computer is presented with example inputs and their desired outputs, given by a "teacher", and the goal is to learn a general rule that maps inputs to outputs.

2. Unsupervised learning: No labels are given to the learning algorithm, leaving it on its own to find structure in its input. Unsupervised learning can be a goal (discovering hidden patterns in data) or a means towards an end (feature learning).

3. Reinforcement learning: A computer program interacts with a dynamic environment in which it must perform a certain goal (such as driving a vehicle or playing a game against an opponent) As it navigates its problem space, the program is provided feedback that's analogous to rewards, which it tries to maximize. However, our project is based on Supervised Learning Algorithms that uses Accuracy, Recall, Precision Score and Confusion Matrix to Predict whether Brain tumor is present or not.

4.3 DEEP LEARNING

Deep learning is a class of machine learning algorithms that uses multiple layers to progressively extract higher level features from the raw input. For example, in image processing, lower layers may identify edges, while higher layers may identify the con relevant to a human such as digits or letters or faces. Most modern deep learning models are based on artificial neural networks, specifically, Convolutional Neural Networks (CNN)s. Deep learning eliminates some of data pre-processing that is typically involved with machine learning. These algorithms can ingest and process unstructured data, like text and images, and it automates feature extraction, removing some of the dependency on human experts.

4.4 ALGORITHMS

4.4.1 Long Short-Term Memory (LSTM) :

LSTMs are a particular type of recurrent neural network (RNN) [33]. They can capture context-specific information from large datasets and use them for future prediction. As its name suggests, each LSTM unit or cell remembers the information for a long or short duration of time. Predicting the output of its new cell state, it takes the information stored in the old cell state. This feature provides memory to the network, which helps improve future predictions. Thus, LSTM networks are best capable of finding out how financial news headlines and closing prices of stocks affect the price trends of several stocks over a more extended period. These networks also decide for how long the past information related to stock price needs to be stored to predict the new price trends accurately.

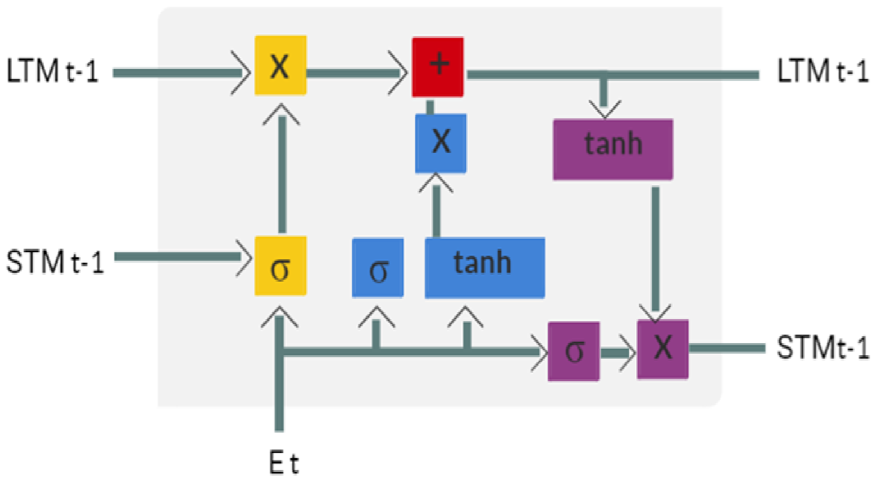

Fig 4.2 : LSTM Cell

4.4.2 ARIMA :

ARIMA is an acronym for “autoregressive integrated moving average.” It’s a model used in statistics and econometrics to measure events that happen over a period of time. The model is used to understand past data or predict future data in a series. It’s used when a metric is recorded in regular intervals, from fractions of a second to daily, weekly or monthly periods. The ARIMA model predicts a given time series based on its own past values. It can be used for any nonseasonal series of numbers that exhibits patterns and is not a series of random events. For example, sales data from a clothing store would be a time series because it was collected over a period of time. One of the key characteristics is the data is collected over a series of constant, regular intervals. A modified version can be created to model predictions over multiple seasons.

An ARIMA model is characterized by 3 terms: p, d, q .Where, p is the order of the AR term, q is the order of the MA term d is the number of differencing required to make the time series stationary.

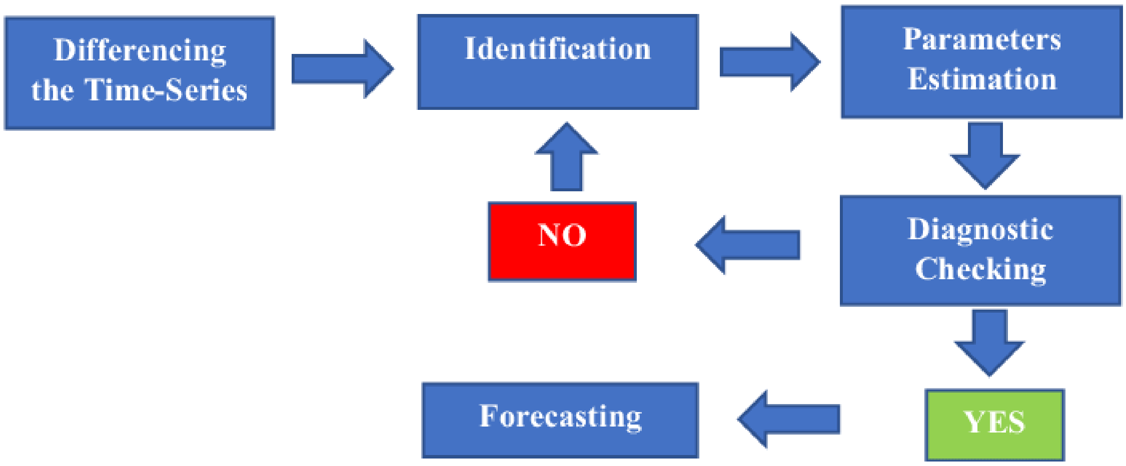

Fig 4.3 : ARIMA Methodology

4.4.3 Linear Regression :

Linear Regression is an ML algorithm used for supervised learning. Linear regression performs the task to predict a dependent variable based on the given independent variable(s). So, this regression technique finds out a linear relationship between a dependent variable and the other given independent variables. In regression set of records are present with X and Y values and this values are used to learn a function, so that if you want to predict Y from an unknown X this learn function can be used. In regression we have to find value of Y, So, a function is required which predicts Y given XY is continuous in case of regression.

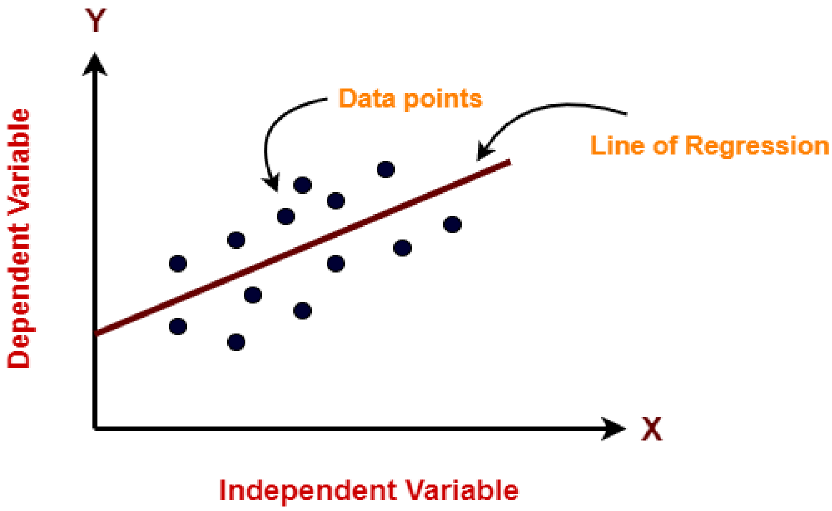

Fig 4.4 : Linear regression Illustration

In the figure above, on X-axis is the independent variable and on Y-axis is the output. The regression line is the best fit line for a model. And our main objective in this algorithm is to find this best fit line.

4.4.4 Rainbow DQN Agent :

Rainbow DQN is an off-policy deep reinforcement learning algorithm that is the state-of-the-art technique in the field of reinforcement learning. Rainbow DQN is used for predicting buy/sell signals on forecasted closing price from the Ensemble Deep learning technique. It is an extended DQN that combines several improvements into a single learner. Specifically: It uses Double Q-Learning to tackle overestimation bias, Prioritized Experience Replay to prioritize important transitions, dueling networks, multi-step learning.

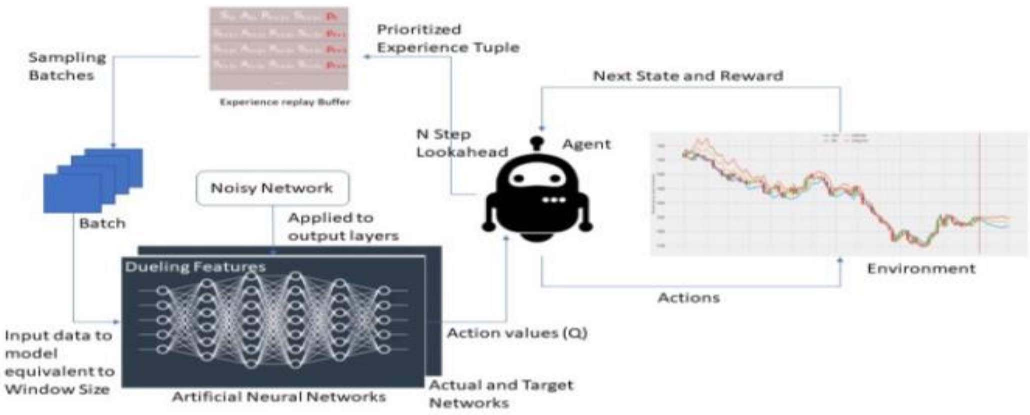

Fig 4.5 : Working of Rainbow DQN

CHAPTER 5

SYSTEM REQUIREMENTS ANALYSIS AND SPECIFICATION

5.1 HARDWARE REQUIREMENTS :

The selection of hardware configuration is a very important task related to the software development. Random Access Memory may affect adversely on the speed and the correspondingly on the efficiency of the entire system. The processor should be powerful to handle of the operations. The hard disk should have sufficient capacity to store the database and the application. The network should be efficient to handle the communication fast.

- Processor : Intel i5/AMD Ryzen5
- Memory : 2 GB.
- Hard Disk : 40 GB.

5.2 SOFTWARE REQUIREMENTS :

- Operating System: Windows 7 and above
- Front End: HTML,CSS
- Backend: Python, SQL
- Platforms: Wordpress, Anaconda spyder, MySQL.

A set of programs associated with the operating operation with a computer is called software. Software is the part of computer system which enables the user to interact with the several physical hardware devices. The minimum software requirement specification for developing this project are as follows:

Presentation	:	Power Point 2020
Documentation Tools	:	MS Word 2020

5.3 LANGUAGES USED :

Python

Python is an interpreted, object-oriented, high-level programming language with dynamic semantics. Its high-level built in data structures, combined with dynamic typing and dynamic binding, make it very attractive for Rapid Application Development, as well as for use as a scripting or glue language to connect existing components together. When it comes to data science, there are some sort of programming language or tool, like Python. Python as a programming language has become very popular in recent times. It has been used in data science, IoT, AI, and other technologies, which has added to its popularity. Python is used as a programming language for data science because it contains costly tools from a mathematical or statistical perspective.

SQL

Structured Query Language (SQL) is a standardized programming language that is used to manage relational databases and perform various operations on the data in them. Initially created in the 1970s, SQL is regularly used not only by database administrators, but also by developers writing data integration scripts and data analysts looking to set up and run analytical queries**.** SQL is used for the following :

- Modifying database table and index structures;
- Adding, Updating and Deleting rows of data; and
- Retrieving subsets of information from within relational database management systems (RDBMSes) -- this information can be used for transaction processing, analytics applications and other applications that require communicating with a relational database.

SQL queries and other operations take the form of commands written as statements and are aggregated into programs that enable users to add, modify or retrieve data from database tables. A table is the most basic unit of a database and consists of rows and columns of data. A single table holds records, and each record is stored in a row of the table. Tables are the most used type of database objects, or structures that hold or reference data in a relational database. Other types of database objects include the following:

- Views are logical representations of data assembled from one or more database tables.
- Indexes are lookup tables that help speed up database lookup functions.
- Reports consist of data retrieved from one or more tables, usually a subset of that data that is selected based on search criteria.

5.4 PLATFORMS USED:

- **Anaconda Spyder :** Spyder is a powerful scientific environment written in Python, for Python, and designed by and for scientists, engineers and data analysts. It features a unique combination of the advanced editing, analysis, debugging and profiling functionality of a comprehensive development tool with the data exploration, interactive execution, deep inspection and beautiful visualization capabilities of a scientific package.There are various IDEs in the market to select from such as Spyder, Atom, Pycharm, Pydev etc. Its interface allows the user to scroll through various data variables and also ready to use online help option. The output of the code can be viewed in the python console on the same screen. One can work on different scripts at a moment and then try them out one by one in the same console or different as per their choice all the variables used will be stored in the variable explorer tab. It also provides an option to view graphs and visualizations in the plot window.

- **WordPress :** WordPress is the simplest, most popular way to create your own website or blog. In fact, WordPress powers over 43.3% of all the websites on the Internet. Yes – more than one in four websites that you visit are likely powered by WordPress. On a slightly more technical level, WordPress is an open-source content management system licensed under GPLv2, which means that anyone can use or modify the WordPress software for free. A content management system is basically a tool that makes it easy to manage important aspects of your website – like content – without needing to know anything about programming. The end result is that WordPress makes building a website accessible to anyone – even people who aren't developers.

- **My SQL :** MySQL is the most widely used open-source database engine in Linux and cloud-based platforms.Almost every web hosting service provider offers a basic MySQL instance included in its web hosting plans at no extra cost.The web+database hosting combo is a preferred option for new or low-traffic websites since the combo frees the system administrator from all the hassles of managing diverse services. But when data management becomes critical in high-volume applications or websites, it could make sense to decouple both services and keep a dedicated hosting just for the database. DBaaS (database as a service) is also a preferred choice if you are setting up the data layer of an application before knowing how you are going to access that data.

Another advantage of having your MySQL installation hosted separately is that you can manage backups, replication, monitoring, and other important features independently from the rest of the hosted services. Also, a managed MySQL platform lets you access it with external tools of your choice, besides the basic tools that the provider offers by default.

5.5 LIBRARIES USED :

- **Pandas :** Pandas is a BSD (Berkeley Software Distribution) licensed open source library. This popular library is widely used in the field of data science. They are primarily used for data analysis, manipulation, cleaning, etc. Pandas is built on top of the NumPy package, meaning a lot of the structure of NumPy is used or replicated in Pandas. Data in pandas is often used to feed statistical analysis in SciPy, plotting functions from Matplotlib, and machine learning algorithms in Scikit-learn.
- **NumPy :** NumPy is one of the most widely used open-source Python libraries, focusing on scientific computation. It features built-in mathematical functions for quick computation and supports big matrices and multidimensional data. Mathematical operations on NumPy's nd array objects are up to 50x faster than iterating over native Python lists using loops. The efficiency gains are primarily due to NumPy storing array elements in an ordered single location within memory, eliminating redundancies by having all elements be the same type and making full use of modern CPUs.
- **Keras :** Keras is a Python-based open-source neural network library that lets us experiment with deep neural networks quickly. With deep learning becoming more common, Keras emerges as a great option because, according to the creators, it is an API (Application Programming Interface) designed for humans, not machines. Keras is a high-level neural networks API, capable of running on top of Tensorflow, Theano, and CNTK. It enables fast experimentation through a high level, user-friendly, modular and extensible API .Keras can also be run on both CPU and GPU. Keras was developed and is maintained by Francois Chollet and is part of the TensorFlow core, which makes it Tensorflows preferred high-level API.
- **TensorFlow :** TensorFlow is a high-performance numerical calculation library that is open source. It is also employed in deep learning algorithms and machine

learning algorithms .TensorFlow is an end-to-end open source platform for machine learning. TensorFlow is a rich system for managing all aspects of a machine learning system; however, this class focuses on using a particular TensorFlow API to develop and train machine learning models.

- **Matplotlib :** This library is responsible for plotting numerical data. And that's why it is used in data analysis. It is also an open-source library and plots high-defined figures like pie charts, histograms, scatterplots, graphs, etc. Matplotlib is one of the most popular Python packages used for data visualization. It is a cross-platform library for making 2D plots from data in arrays .Matplotlib is written in Python and makes use of NumPy, the numerical mathematics extension of Python. It provides an object-oriented API that helps in embedding plots in applications using Python GUI toolkits such as PyQt, WxPythonotTkinter. It can be used in Python and IPython shells, Jupyter notebook and web application servers also. Matplotlib along with NumPy can be considered as the open source equivalent of MATLAB.
- **Streamlit :** Streamlit is a free and open-source framework to rapidly build and share beautiful machine learning and data science web apps. It is a Python-based library specifically designed for machine learning engineers. Data scientists or machine learning engineers are not web developers and they're not interested in spending weeks learning to use these frameworks to build web apps. Instead, they want a tool that is easier to learn and to use, as long as it can display data and collect needed parameters for modeling. Streamlit allows you to create a stunning-looking application with only a few lines of code.
- **Flask :** Flask is a web application framework written in Python. It is developed by Armin Ronacher, who leads an international group of Python enthusiasts named Pocco. Flask is based on the Werkzeug WSGI toolkit and Jinja2 template engine. Both are Pocco projects. Flask is often referred to as a micro framework. It aims to keep the core of an application simple yet extensible. Flask does not have built-in abstraction layer for database handling, nor does it have form a validation support. Instead, Flask supports the extensions to add such functionality to the application.
- **Yfinance :** yfinance is a popular open source library developed by Ran Aroussi as a means to access the financial data available on Yahoo Finance. Yahoo Finance offers an excellent range of market data on stocks, bonds, currencies and cryptocurrencies. It also offers market news, reports and analysis and additionally

options and fundamentals data- setting it apart from some of it's competitors. yfinance is highly Pythonic in it's design and incredibly streamlined. It's as easy as creating a ticker object for a particular ticker/list of tickers and then just calling all the methods on this object.

- **Alpha vantage :** Alpha Vantage provides enterprise-grade financial market data through a set of powerful and developer-friendly data APIs and spreadsheets. From traditional asset classes (e.g., stocks, ETFs, mutual funds) to economic indicators, from foreign exchange rates to commodities, from fundamental data to technical indicators, Alpha Vantage is your one-stop-shop for real-time and historical global market data delivered through cloud-based APIs, Excel, and Google Sheets. The Alpha Vantage Stock API provides free JSON access to the stock market, plus a comprehensive set of technical indicators.
- **NLTK :** NLTK is a leading platform for building Python programs to work with human language data. It provides easy-to-use interfaces to over 50 corpora and lexical resources such as WordNet, along with a suite of text processing libraries for classification, tokenization, stemming, tagging, parsing, and semantic reasoning, wrappers for industrial-strength NLP libraries, and an active discussion forum. Natural Language Processing with Python provides a practical introduction to programming for language processing.It guides the reader through the fundamentals of writing Python programs, working with corpora, categorizing text, analysing linguistic structure, and more.
- **Scikit-learn :** The name "SciPy" stands for "Scientific Python". It is an open-source library used for high-level scientific computations. This library is built over an extension of Numpy. It works with Numpy to handle complex computations. It was originally called scikits.learn and was initially developed by David Cournapeau as a Google summer of code project in 2007.

CHAPTER 6

SYSTEM DESIGN

The models will be trained and tested on data that will be collected from National Stock Exchange, India website which has prices structured in 1day intervals and from Alpha Vantage a finance API that provides intraday data structured in 1-minute, 5-minute, 15-minute intervals. Data like news headlines will be fetched from various business websites for sentiment analysis. We will consider multiple news headlines being fetched every minute on a real-time basis.

The deep learning models used are LSTM, GRU, ARIMA, Linear Regression. LSTM will be trained on historical data; sentiment score will be calculated from news headlines using an encoder network and GRU will be trained on it, such training is then applied to inference for forecasting. The Ensemble will be done by calculating the average output from both the modules.

6.1 ER DIAGRAM

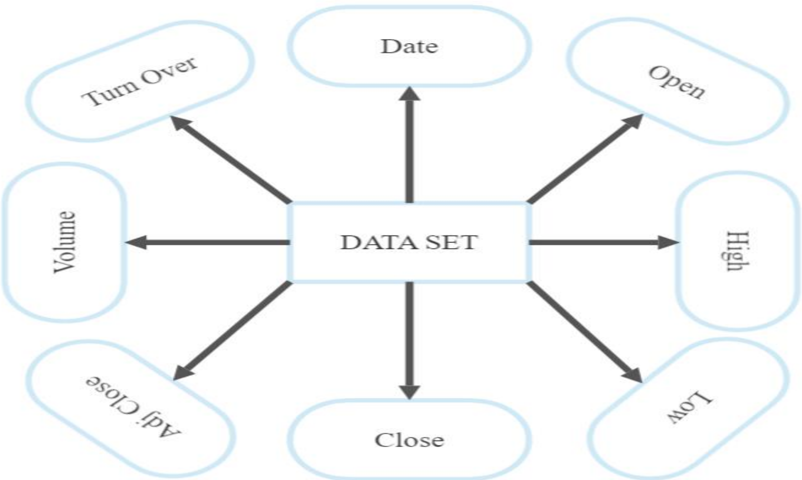

Fig 6.1 : ER Diagram of the Dataset

	A	B	C	D	E	F	G
1	Date	Open	High	Low	Close	Adj Close	Volume
2	28-04-2021	134.31	135.02	133.08	133.58	131.9912	107760100
3	29-04-2021	136.47	137.07	132.45	133.48	131.8924	151101000
4	30-04-2021	131.78	133.56	131.07	131.46	129.8964	109839500
5	03-05-2021	132.04	134.07	131.83	132.54	130.9635	75135100
6	04-05-2021	131.19	131.49	126.7	127.85	126.3293	137564700
7	05-05-2021	129.2	130.45	127.97	128.1	126.5764	84000900
8	06-05-2021	127.89	129.75	127.13	129.74	128.1969	78128300
9	07-05-2021	130.85	131.26	129.48	130.21	128.8798	78973300
10	10-05-2021	129.41	129.54	126.81	126.85	125.5541	88071200

Fig 6.2 : Dataset

6.2 SYSTEM DESIGN

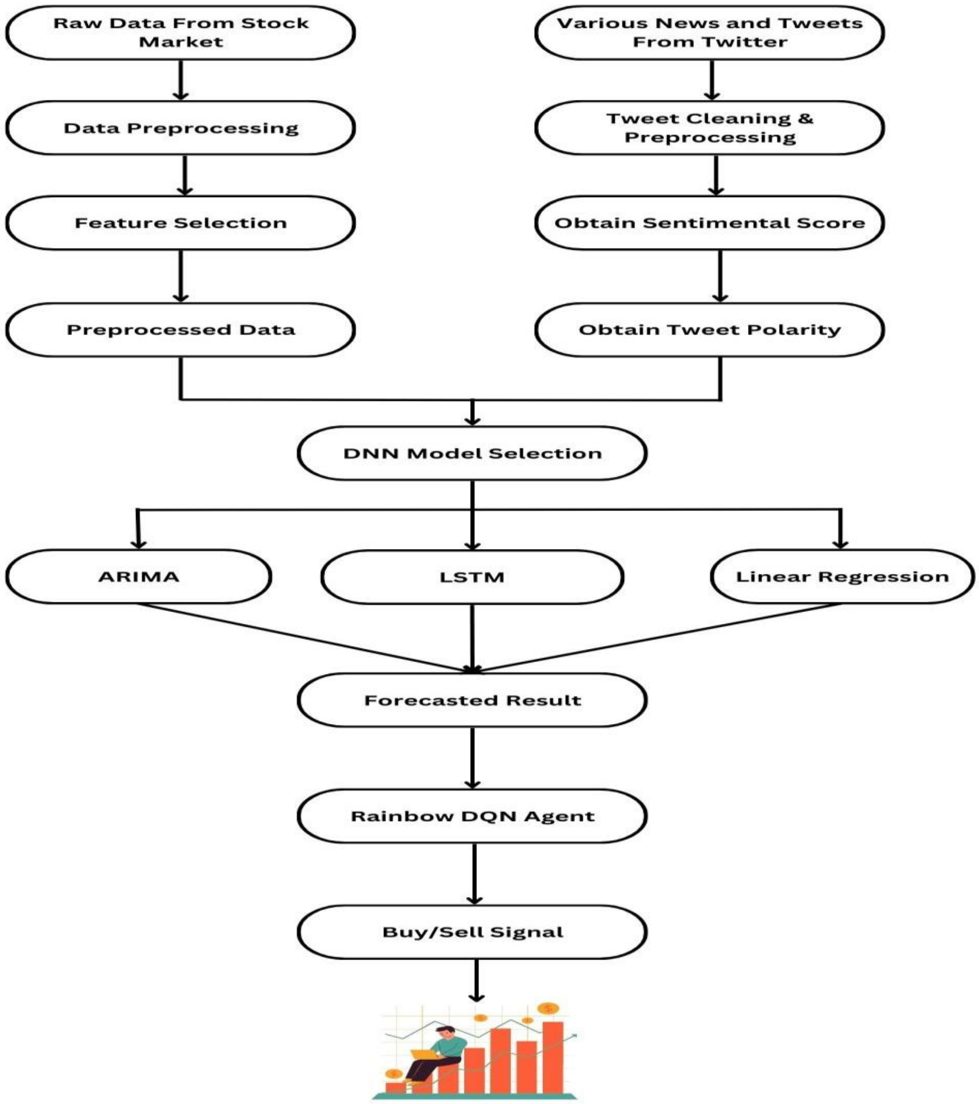

Fig 6.3 : System Flow Diagram

The historical raw data from stock market is extracted using Alpha Vantage API and yahoo finance library as well as the news and tweets from twitter are extracted from tweepy library. Next the raw data from both stock market and twitter is cleaned and processed using the numpy, pandas and textblob library. Now from the preprocessed data the necessary features i,e open, close, high, low, volume, adj close will be selected for model and from cleaned and tokenized tweets the sentimental score is obtained. Then from sentimental score the tweet polarity that is positive, neutral or negative is obtained. Now the preprocessed data from stock market and tweet polarity is fed into deep learning models such as LSTM, ARIMA, Linear Regression algorithm these models gives the forecasted stock closing price for next seven days then given as an input to RNN Model that is then trained on sentiment score and historical data to forecast price, average of results from these models is calculated. Next the Rainbow DQN agent takes the tweet polarity and forecasted result values as well as the sentimental score from the above steps and gives the recommendation of the selected stock then with the help of this model it gives the buy/sell signal for a particular stock.

6.3 SENTIMENTAL ANALYSIS ARCHITECTURE

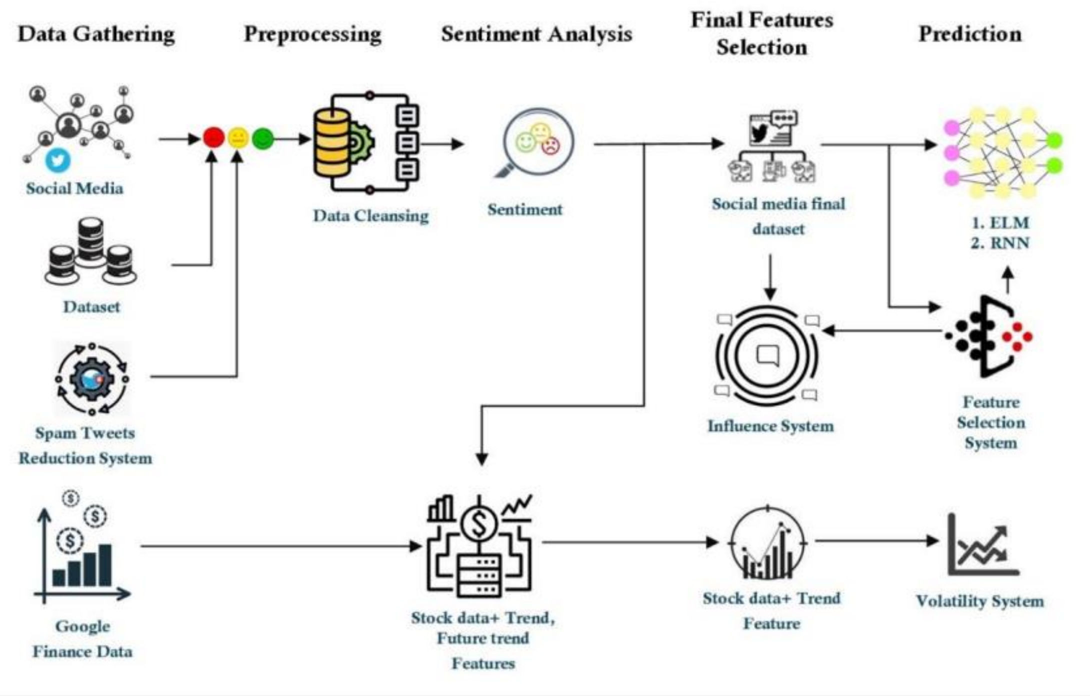

Fig 6.4 : Working Architecture of Sentimental Analysis Module

In this module the raw data is gathered from twitter using the tweepy package then this data is preprocessed where data cleaning is done to remove the white spaces and non-ASCII characters next sentimental analysis is done from the tokenized tweets and news to obtain sentimental score. Next from the analysed result and sentimental score the final features are selected to obtain the tweet polarity such as positive, neutral, negative. Finally from the obtained tweet polarity from the previous step the fundamental prediction of stocks based on tweets is taken place to get the final result.

CHAPTER 7

SYSTEM IMPLEMENTATION AND TESTING

7.1 MODULES DESCRIPTION

The system can be divided into Five modules.

- **Module 1**: Data Collection and Preprocessing
- **Module 2**: Deep Learning Module
- **Module 3**: Sentimental Analysis Module
- **Module 4**: Rainbow DQN Module
- **Module 5**: Implementing the Model in Flask

7.1.1 Data Collection :

Collecting data for training the ML model is the basic step in the machine learning pipeline. The predictions made by ML systems can only be as good as the data on which they have been trained. Following are some of the problems that can arise in data collection:

- Inaccurate data: The collected data could be unrelated to the problem statement.
- Missing data: Sub-data could be missing. That could take the form of empty values in columns or missing images for some class of prediction.

Real-world raw data and images are often incomplete, inconsistent and lacking in certain behaviors or trends. They are also likely to contain many errors. So, once collected, they are pre-processed into a format the machine learning algorithm can use for the model. Pre-processing includes a number of techniques and actions :

- Data cleaning: These techniques, manual and automated, remove data incorrectly added or classified.
- Data imputations: Most ML frameworks include methods and APIs for balancing or filling in missing data. Techniques generally include imputing missing values with standard deviation, mean, median
- Data integration: Combining multiple datasets to get a large corpus can overcome incompleteness in a single dataset.

7.1.2 Deep Learning Module :

The purpose of this module is to output Stock Prediction value. Stock Prediction value is the strength of difference in opening price and closing price. For this we need to predict the closing price of the stock. This is achieved by applying Machine Learning on Historical data of the stock. The maximum number of features required to accurately predict a stock's closing price for a specific day are given as follows.

1. Opening price of prediction day
2. Lowest and highest prices of the prediction day
3. Simple Moving Average
4. Exponential moving average of opening and closing prices of the prediction day
5. Exponential moving average of lowest and highest prices of the prediction day
6. Bollinger Bands of opening and closing prices of the prediction day
7. Bollinger Bands of lowest and highest prices of the prediction day

The training data is then fitted by a machine learning module and is used to predict the closing price of testing data through supervised learning. There are many regressors available scikit learn library. Their accuracy was measured in terms of percentage error rate and Root mean square error(RMSE).

7.1.3 Sentimental Analysis Module :

The purpose of this module is to obtain the sentiment value of latest news headlines regarding each stock and output its average as sentiment value to fuzzy module. The steps used in this module are as follows :

1. Data Collection: The data is collected by crawling through Indian Financial news website www.moneycontrol.com and Twitter Minimum 4 news Headlines are scraped for each stock and stored against the company Symbol.

2. Tokenizing: Each news headline is broken down into sentences and then in turn broken down into words.

3. Lemmatizing: It is the process of reducing inflected (or sometimes derived) words to their word stem, base or root form. For example, "the boy's cars are different colours" reduces to "the boy car be differ colour".

4. Finding Most Informative Features: Words that contribute most in adding polarity to a sentence are found.

5. Classifying features into positive and negative: These are then classified into positive and negative using nltk packages.

6. Adding these features to the sentiment analyser lexicon: These words are then added to the sentiment analyser wordlist with appropriate strength for positive and negative words.

7. Classifying the testing data into positive and negative sentiments using training set: Now our sentiment analyser is ready for classifying financial news from our sources.

7.1.4 Rainbow DQN Module :

Rainbow DQN Agent was trained and tested on data from GOOG 2015 - GOOG 2023. The data was divided into train-validation-test with 80% - 10% - 10% split and given to the agent for training. The results of all the descendants of Rainbow DQN were compared with Rainbow DQN. The comparison was based on return of investment each agent could get when tested on the testing split of GOOG which is equivalent to 1 year of data, reward curve with respect to episodes. The agents were trained for 10-30 epochs, before or after which signs of underfitting and overfitting were observed during training. The agents were saved after every 10 epochs regardless of the reward it got at that epoch. The rewards changed significantly moving into positive as a sign of good action values and negative as a sign of punishment to the agent. The Agent can trade only 1 item at a time in our present implementation. Window Size of 10 and Batch Size of 32 were considered for our implementation. Here, window size refers to a sliding window on the data that considers 10 data inputs at a time which is given to the model as state size. Adam Optimizer was used for optimizing gradient descent of the model. The output size/action size of 3 is obvious due tot the fact that only 3 actions can be performed that is buy, hold, sell. Since, the agent was trained for 25 epochs, number of episodes referred to 30 times the length of the test data. The Agent was then finally given output of ensemble algorithms as test input to check its performance

7.1.5 Implementing the Model in Flask :

Finally, the model is implemented using a Flask where the yahoo finance library fetches the data of given stock using the alpha vantage key. Then, the preprocessing function is called to get the data from the API and render the data set to required columns only. Then data is converted using numpy and pandas and finally sent to machine learning models for forecasting. Meanwhile the data across web especially twitter is fetched through tweepy and text blob and preprocessed using tweet processor of anaconda package and recommends the opinion on the given stock.

7.2 SYSTEM CODE

The proposed system is implemented using the following code to provide web interface to users to predict the stock.

7.2.1 Functions To Fetch Data :

```
def get_historical(quote):
    end = datetime.now()
    start = datetime(end.year-2,end.month,end.day)
    data = yf.download(quote, start=start, end=end)
    df = pd.DataFrame(data=data)
    df.to_csv(''+quote+'.csv')
    if(df.empty):
        ts = TimeSeries(key='N6A6QT6IBFJOPJ70',output_format='pandas')
        data, meta_data = ts.get_daily_adjusted(symbol='NSE:'+quote, outputsize='full')
        #Format df
        #Last 2 yrs rows => 502, in ascending order => ::-1
        data=data.head(503).iloc[::-1]
        data=data.reset_index()
        #Keep Required cols only
        df=pd.DataFrame()
        df['Date']=data['date']
        df['Open']=data['1. open']
        df['High']=data['2. high']
        df['Low']=data['3. low']
        df['Close']=data['4. close']
        df['Adj Close']=data['5. adjusted close']
        df['Volume']=data['6. volume']
        df.to_csv(''+quote+'.csv',index=False)
    return
```

7.2.2 Preprocessing :

```
df = pd.read_csv(''+quote+'.csv')
print("##############################################################################")
    print("Today's",quote,"Stock Data: ")
```

```
        today_stock=df.iloc[-1:]
        print(today_stock)
    print("##############################################################################")
        df = df.dropna()
        code_list=[]
        for i in range(0,len(df)):
            code_list.append(quote)
        df2=pd.DataFrame(code_list,columns=['Code'])
        df2 = pd.concat([df2, df], axis=1)
        df=df2
        arima_pred, error_arima=ARIMA_ALGO(df)
        lstm_pred, error_lstm=LSTM_ALGO(df)
        df, lr_pred, forecast_set,mean,error_lr=LIN_REG_ALGO(df)
        polarity,tw_list,tw_pol,pos,neg,neutral = retrieving_tweets_polarity(quote)
        idea, decision=recommending(df, polarity,today_stock,mean)
        print()
        print("Forecasted Prices for Next 7 days:")
        print(forecast_set)
        today_stock=today_stock.round(2)
        return
render_template('results.html',quote=quote,arima_pred=round(arima_pred,2),lstm_pred=roun
d(lstm_pred,2),lr_pred=round(lr_pred,2),open_s=today_stock['Open'].to_string(index=False),
close_s=today_stock['Close'].to_string(index=False),adj_close=today_stock['Adj Close'].
to_string(index=False),tw_list=tw_list,tw_pol=tw_pol,idea=idea,decision=decision,high_s=
today_stock['High'].to_string(index=False)low_s=today_stock['Low'].to_string(index=False),
vol=today_stock['Volume'].to_string(index=False),forecast_set=forecast_set,error_lr=round(
error_lr,2),error_lstm=round(error_lstm,2),error_arima=round(error_arima,2))
if __name__ == '__main__':
  app.run()
```

7.2.3 LSTM Section :

```
def LSTM_ALGO(df):
        #Split data into training set and test set
```

```
dataset_train=df.iloc[0:int(0.8*len(df)),:]
dataset_test=df.iloc[int(0.8*len(df)):,:]
#TO PREDICT STOCK PRICES OF NEXT N DAYS, STORE PREVIOUS N DAYS IN MEMORY WHILE TRAINING
# HERE N=7
dataset_train=pd.read_csv('Google_Stock_Price_Train.csv')
training_set=df.iloc[:,4:5].values# 1:2, to store as numpy array else Series obj will be store
#Feature Scaling
from sklearn.preprocessing import MinMaxScaler
sc=MinMaxScaler(feature_range=(0,1))#Scaled values btween 0,1
training_set_scaled=sc.fit_transform(training_set)
#In scaling, fit_transform for training, transform for test
#Creating data stucture with 7 timesteps and 1 output.
#7 timesteps meaning storing trends from 7 days before current day to predict next output
X_train=[] #memory with 7 days from day i
y_train=[] #day i
for i in range(7,len(training_set_scaled)):
    X_train.append(training_set_scaled[i-7:i,0])
    y_train.append(training_set_scaled[i,0])
#Convert list to numpy arrays
X_train=np.array(X_train)
y_train=np.array(y_train)
X_forecast=np.array(X_train[-1,1:])
X_forecast=np.append(X_forecast,y_train[-1])
#Reshaping: Adding 3rd dimension
X_train=np.reshape(X_train,(X_train.shape[0],X_train.shape[1],1))#.shape 0=row,1=col
X_forecast=np.reshape(X_forecast, (1,X_forecast.shape[0],1))
```

7.2.4 Building RNN :

```
from keras.models import Sequential
from keras.layers import Dense
from keras.layers import Dropout
from keras.layers import LSTM
```

```
#Initialise RNN
regressor=Sequential()
#Add first LSTM layer
regressor.add(LSTM(units=50,return_sequences=True,input_shape=(X_train.shape[1],1)))
#units=no. of neurons in layer
#input_shape=(timesteps,no. of cols/features)
#return_seq=True for sending recc memory. For last layer,
retrun_seq=False since end of the line
regressor.add(Dropout(0.1))
#Add 2nd LSTM layer
regressor.add(LSTM(units=50,return_sequences=True))
regressor.add(Dropout(0.1))
#Add 3rd LSTM layer
regressor.add(LSTM(units=50,return_sequences=True))
regressor.add(Dropout(0.1))
#Add 4th LSTM layer
regressor.add(LSTM(units=50))
regressor.add(Dropout(0.1))
#Add o/p layer
regressor.add(Dense(units=1))
#Compile
regressor.compile(optimizer='adam',loss='mean_squared_error')
#Training
regressor.fit(X_train,y_train,epochs=25,batch_size=32 )
#For lstm, batch_size=power of 2
#Testing
dataset_test=pd.read_csv('Google_Stock_Price_Test.csv')
real_stock_price=dataset_test.iloc[:,4:5].values
#To predict, we need stock prices of 7 days before the test set
#So combine train and test set to get the entire data set
dataset_total=pd.concat((dataset_train['Close'],dataset_test['Close']),axis=0)
testing_set=dataset_total[ len(dataset_total) -len(dataset_test) -7: ].values
testing_set=testing_set.reshape(-1,1)
#-1=till last row, (-1,1)=>(80,1). otherwise only (80,0)
```

```
#Feature scaling
testing_set=sc.transform(testing_set)
#Create data structure
X_test=[]
for i in range(7,len(testing_set)):
X_test.append(testing_set[i-7:i,0])
#Convert list to numpy arrays
X_test=np.array(X_test)
#Reshaping: Adding 3rd dimension
X_test=np.reshape(X_test, (X_test.shape[0],X_test.shape[1],1))
#Testing Prediction
predicted_stock_price=regressor.predict(X_test)
#Getting original prices back from scaled values
predicted_stock_price=sc.inverse_transform(predicted_stock_price)
fig = plt.figure(figsize=(7.2,4.8),dpi=65)
plt.plot(real_stock_price,label='Actual Price')
plt.plot(predicted_stock_price,label='Predicted Price')
plt.legend(loc=4)
plt.savefig('static/LSTM.png')
plt.close(fig)
error_lstm = math.sqrt(mean_squared_error(real_stock_price, predicted_stock_price))
```

7.2.5 Forecasting Prediction :

```
forecasted_stock_price=regressor.predict(X_forecast)
#Getting original prices back from scaled values
forecasted_stock_price=sc.inverse_transform(forecasted_stock_price)
lstm_pred=forecasted_stock_price[0,0]
print()
print("##############################################################################")
print("Tomorrow's ",quote," Closing Price Prediction by LSTM: ",lstm_pred)
print("LSTM RMSE:",error_lstm)
print("#############################################################################")
return lstm_pred,error_lstm
```

7.2.6 Sentiment Analysis :

```
def retrieving_tweets_polarity(symbol):
        stock_ticker_map = pd.read_csv('Yahoo-Finance-Ticker-Symbols.csv')
        stock_full_form = stock_ticker_map[stock_ticker_map['Ticker']==symbol]
        symbol = stock_full_form['Name'].to_list()[0][0:12]
        auth = tweepy.OAuthHandler(ct.consumer_key, ct.consumer_secret)
        auth.set_access_token(ct.access_token, ct.access_token_secret)
user=tweepy.API(auth)
tweets=tweepy.Cursor(user.search,q=symbol,tweet_mode='extended',lang='en',exclude_repli
es=True).items(ct.num_of_tweets)
        tweet_list = [] #List of tweets alongside polarity
        global_polarity = 0 #Polarity of all tweets === Sum of polarities of individual tweets
        tw_list=[] #List of tweets only => to be displayed on web page
        #Count Positive, Negative to plot pie chart
        pos=0  #Num of pos tweets
        neg=1 #Num of negative tweets
        for tweet in tweets:
            count=20 #Num of tweets to be displayed on web page
            #Convert to Textblob format for assigning polarity
            tw2 = tweet.full_text
            tw = tweet.full_text
            #Clean
            tw=p.clean(tw)
            print("-------------------------------CLEANED TWEET-----------------------------")
            print(tw)
            #Replace &amp; by &
            tw=re.sub('&amp;','&',tw)
            #Remove :
            tw=re.sub(':','',tw)
            print("---------------TWEET AFTER REGEX MATCHING------------------------------")
            print(tw)
            #Remove Emojis and Hindi Characters
            tw=tw.encode('ascii', 'ignore').decode('ascii')
```

```
        print("------------TWEET AFTER REMOVING NON ASCII CHARS-----------------")
        print(tw)
        blob = TextBlob(tw)
        polarity = 0 #Polarity of single individual tweet
        for sentence in blob.sentences:
            polarity += sentence.sentiment.polarity
            if polarity>0:
                pos=pos+1
            if polarity<0:
                neg=neg+1
            global_polarity += sentence.sentiment.polarity
        if count > 0:
            tw_list.append(tw2)
        tweet_list.append(Tweet(tw, polarity))
        count=count-1
    if len(tweet_list) != 0:
        global_polarity = global_polarity / len(tweet_list)
    else:
        global_polarity = global_polarity
    neutral=ct.num_of_tweets-pos-neg
    if neutral<0:
        neg=neg+neutral
        neutral=20
    print()
print("#######################################################################")
    print("Positive Tweets :",pos,"Negative Tweets :",neg,"Neutral Tweets :",neutral)
print("#######################################################################")
    labels=['Positive','Negative','Neutral']
    sizes = [pos,neg,neutral]
    explode = (0, 0, 0)
    fig = plt.figure(figsize=(7.2,4.8),dpi=65)
    fig1, ax1 = plt.subplots(figsize=(7.2,4.8),dpi=65)
    ax1.pie(sizes, explode=explode, labels=labels, autopct='%1.1f%%', startangle=90)
    # Equal aspect ratio ensures that pie is drawn as a circle
```

```
        ax1.axis('equal')
        plt.tight_layout()
        plt.savefig('static/SA.png')
        plt.close(fig)
        #plt.show()
        if global_polarity>0:
            print()
print("##########################################################################")
            print("Tweets Polarity: Overall Positive")
print("##########################################################################")
            tw_pol="Overall Positive"
        else:
            print()
print("##########################################################################")
            print("Tweets Polarity: Overall Negative")
print("##########################################################################")
            tw_pol="Overall Negative"
        return global_polarity,tw_list,tw_pol,pos,neg,neutral
```

7.2.7 Recommending Stock Using DQN :

```
def recommending(df, global_polarity,today_stock,mean):
        if today_stock.iloc[-1]['Close'] < mean:
            if global_polarity > 0:
                idea="RISE"
                decision="BUY"
                print()
print("##########################################################################")
 print("According to the ML Predictions and Sentiment Analysis of Tweets, a",idea,"in",quote,
"stock is expected => ",decision)
            elif global_polarity <= 0:
                idea="FALL"
                decision="SELL"
                print()
```

```
print("#######################################################################")
print("According to the ML Predictions and Sentiment Analysis of Tweets, a",idea,"in",
quote,"stock is expected => ",decision)
        else:
            idea="FALL"
            decision="SELL"
            print()
print("#######################################################################")
            print("According to the ML Predictions and Sentiment Analysis of Tweets,
a",idea,"in",quote,"stock is expected => ",decision)
        return idea, decision
```

7.3 SYSTEM TESTING

System testing aims in evaluating an attribute or capability of a program or system and determining that it meets its required results. The purpose of testing can be quality assurance, verification and validation, or reliability estimation. System testing is nothing but subjecting a piece of code to both, controlled as well as uncontrolled operating conditions, to observe the output and examine whether it is in accordance with certain pre-specified conditions.

7.3.1 Test Environment :

- CPU Speed: 1.10 GHz
- RAM: 4 GB
- Hard Disk Capacity: 1 TB
- OS Used: Microsoft Windows 10

7.3.2 Unit Testing :

The essential or most common degree of testing is called unit testing. In unit testing every unit of a program is checked individually. Here, all the models are tested and we have found out that these modules were self-independent and was able to yield the required result.

7.3.3 Integration Testing :

The going with level of testing is known as the Integration Testing. For checking the interface between the front-end and back-end development, integration testing is used. By this testing we can get the correct output. It is observed that our model passed this test.

7.3.4 Functional Testing :

Functional testing is a quality assurance (QA) process and a type of black-box testing that bases its test cases on the specifications of the software component under test. Functions are tested by feeding them input and examining the output, and internal program structure is rarely considered (unlike white-box testing). Functional testing is conducted to evaluate the compliance of a system or component with specified functional requirements. Functional testing usually describes what the system does. It is a case structure strategy where the system is tried against the pleasing necessities/points of interest. Limits are attempted by dealing with the data and looking at the result. Helpful testing ensures that the necessities are suitably satisfied by the application.

CHAPTER 8

RESULTS AND DISCUSSIONS

Fig 8.1 : Home page

The above figure shows home page where new users can register and existing users can login to account .

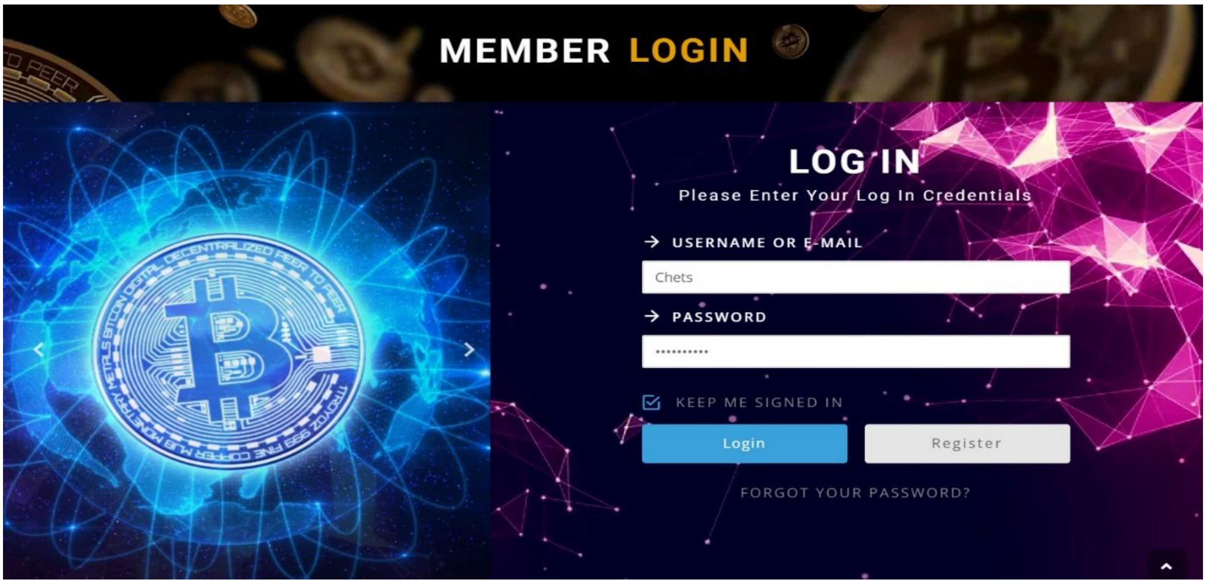

Fig 8.2 : Login page

The above figure shows the login page for existing users where user can login into the account by using the username and password.

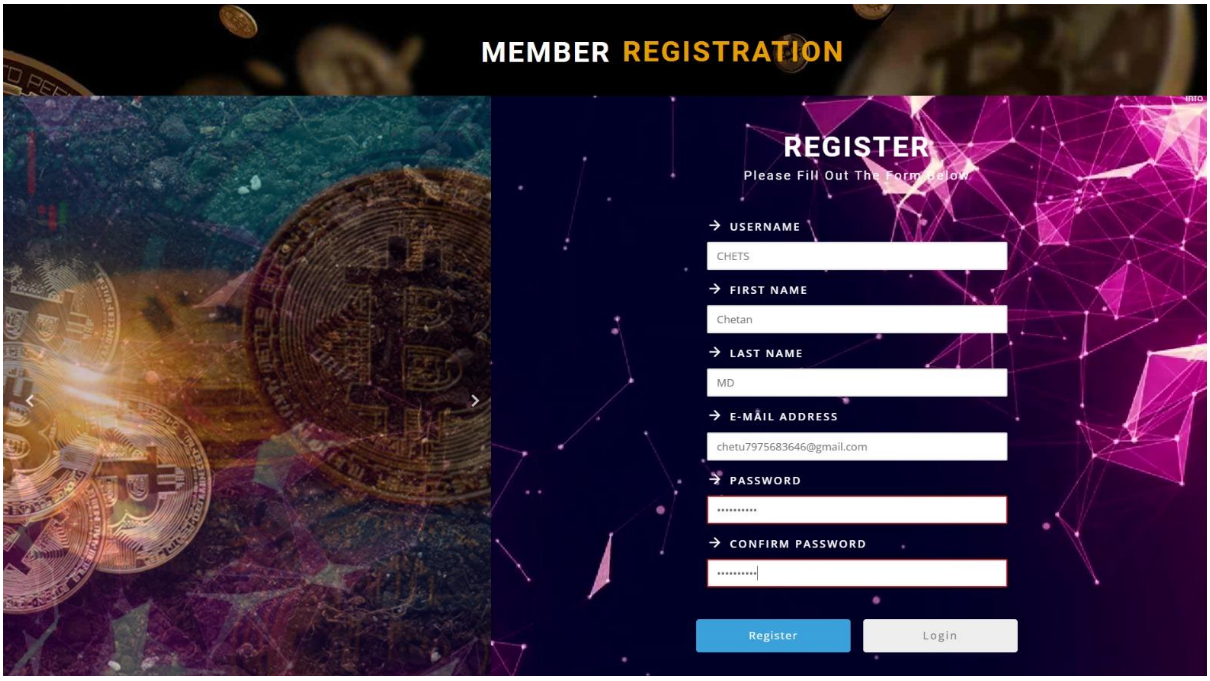

Fig 8.3 : Member Registration Page

The above figure shows the registration page for new users where they register themselves to use the portal .

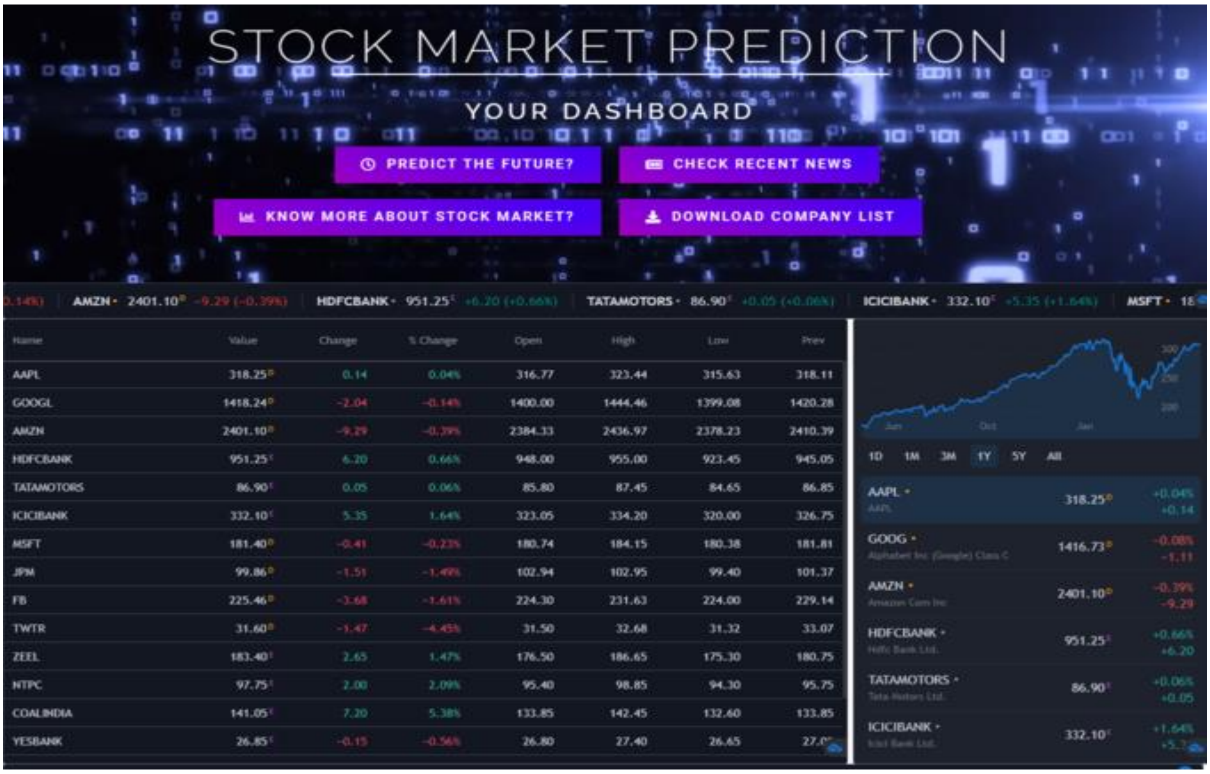

Fig 8.4 : Dashboard of User

The above figure shows the dashboard of users where user can see real time stock market movements.

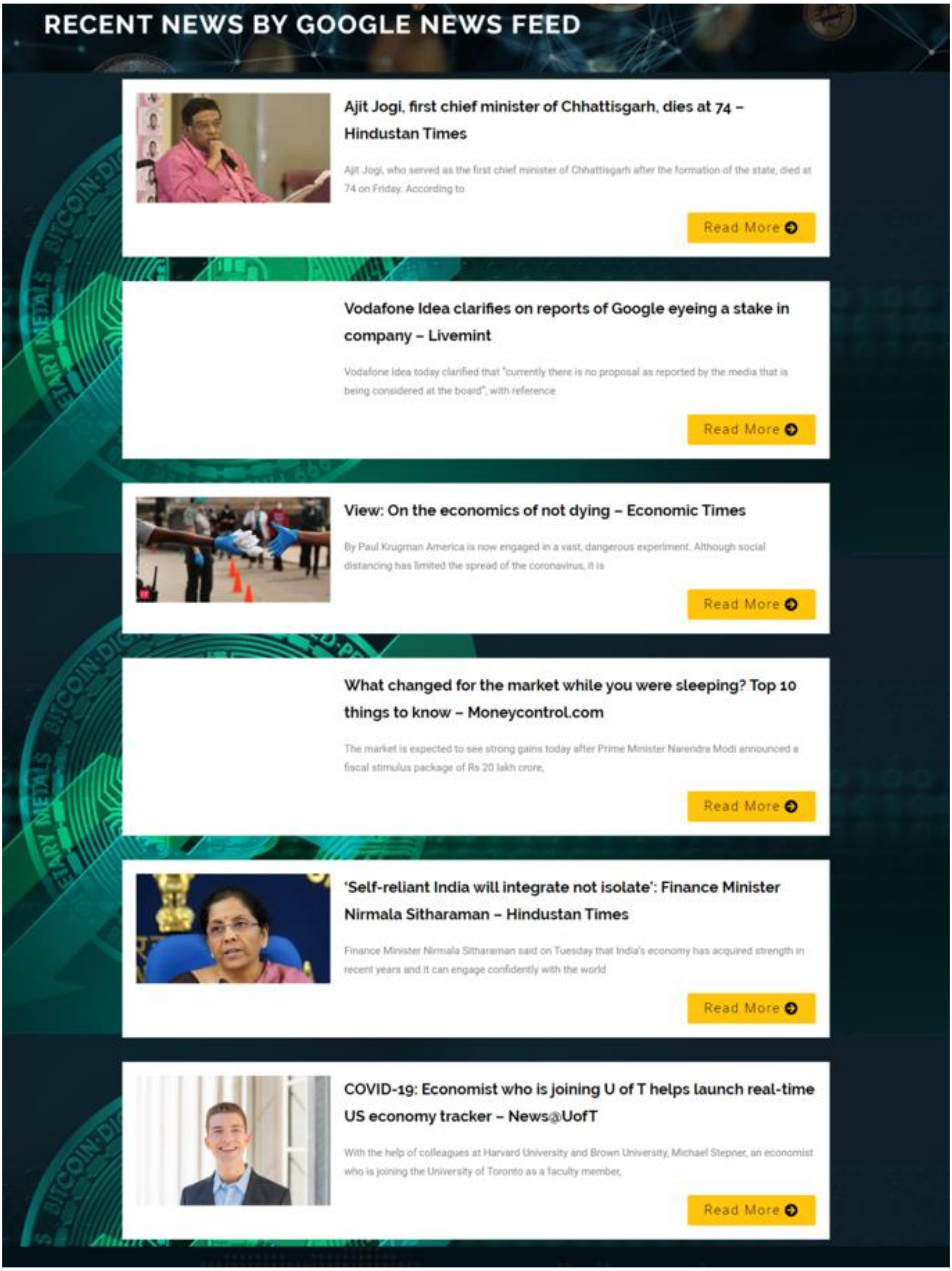

Fig 8.5 : Recent News Page about Stocks

The above figure Shows the recent news which can influence the stock market and trends of investors which helps the investors(users) to get insight of stock market for fundamental analysis.

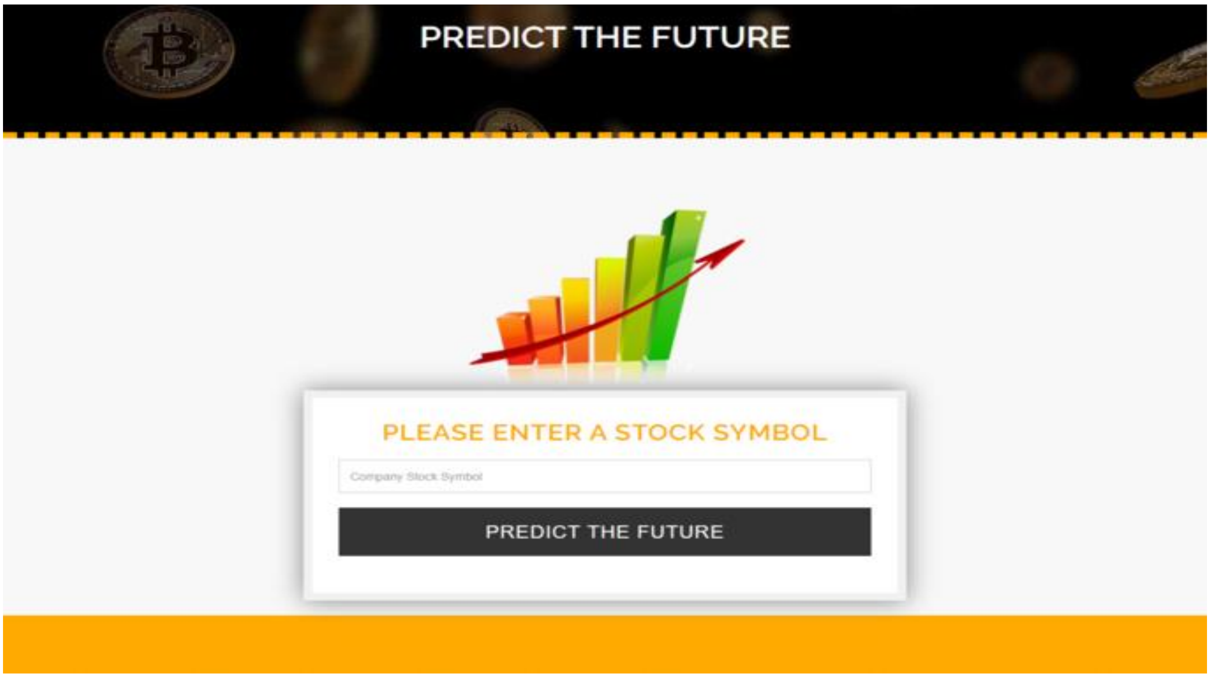

Fig 8.6 : Prediction Page

The above figure shows the prediction page to forecast the stock where user need to give stock symbol as input and can see the prediction of that particular stock.

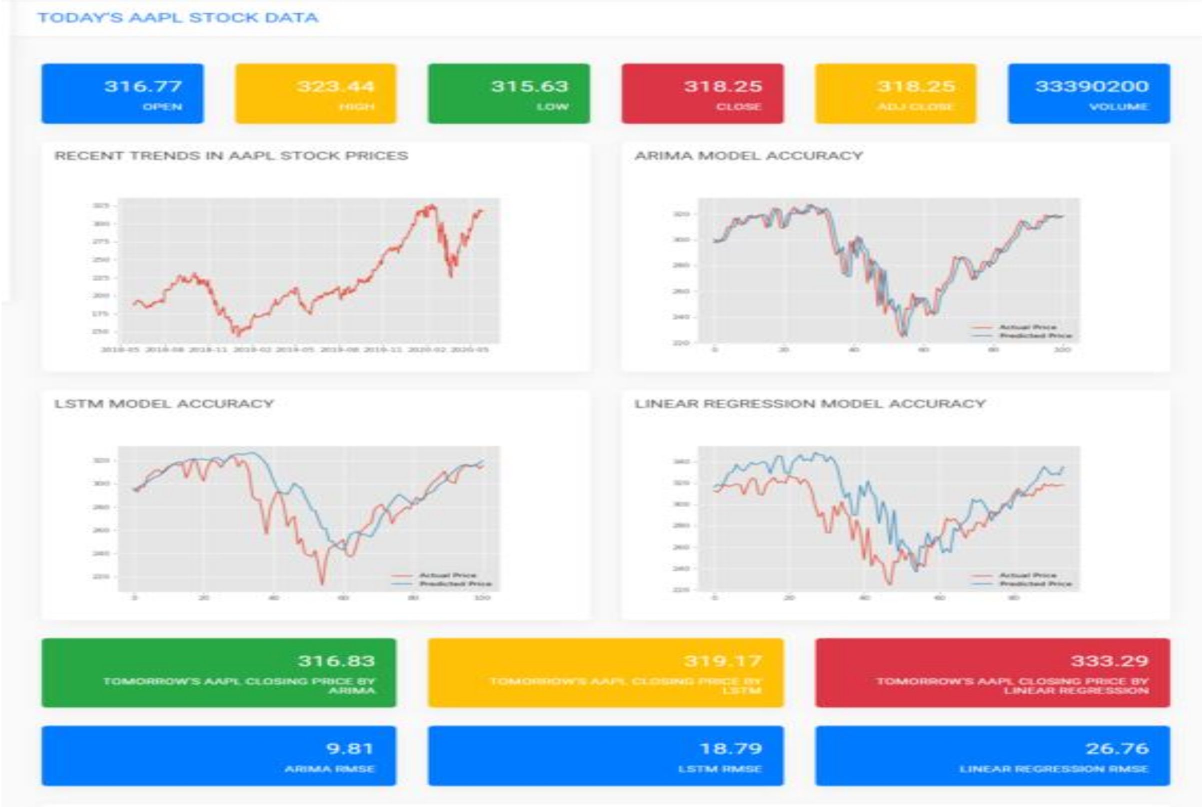

Fig 8.7 : Results Page

The above figure shows the predicted results of the stock along with four machine learning models graphs and next day closing price of the stock.

Fig 8.8 : Overall Results Page

The above figure shows the recent tweets about selected stock and sentimental analysis of selected stock based the tweets and forecasted price of the stock for next seven days.

CONCLUSION AND FUTURE SCOPE

Conclusion :

Predicting stock market returns is a challenging task due to consistently changing stock values which are dependent on multiple parameters which form complex patterns. The historical dataset available on company’s website consists of only few features like high, low, open, close, adjacent close value of stock prices, volume of shares traded etc., which are not sufficient enough. To obtain higher accuracy in the predicted price value new variables have been created using the existing variables. RNN is used for predicting the next day closing price of the stock and for a comparative analysis. Indeed, deep learning algorithms (RNN and LSTM) were our superior models in both approaches. Using Sentiment analysis on the tweets collected from Twitter API and with the closing values of stocks, We created a system which can forecast the stock price movement, getting the financial news by web srcapping and building a web interface to access the Predicted Information. We developed the system to use real-time data and use it to train the model and updating the model using stock dataapisor web scraping stock data. As opposed to a ground survey that would have been conducted otherwise to gauge public sentiment, using Machine Learning techniques for prediction is less costly.

Future Scope :

Despite such results, the model was not able to work well with high volatility and hence value accuracy. More parameters can be considered or more models can be ensembled for more accurate results. Mathematical models like Fast Fourier Transform and Decomposition . The proposed final architecture can also be used to make trading bots. The stock market prediction has been improving gradually and future advancements in technology can give previously obscure and better results. An LSTM base model might work better than a normal Recurrent Neural Network with Rainbow DQN algorithm. Distributional RL was not a part of our implementation and it can significantly improve the present results of our agent.

REFERENCES

[1] M. Usmani; S. H. Adil; K. Raza; S. S. A. Ali, “Stock market prediction wuing machine learning techniques” Kuala Lumpur, Malaysia, 15 December 2018, IEEE.

[2] A. Shakva ; A. Pokhrel ; A. Bhattarai ; P. Sitikhu ; S. Shakva,”Real-Time Stock Market Prediction using neural networks”, Noida, India, 23 August 2018,IEEE.

[3] https://www.alphavantage.co/documentation/ [Accessed: 14 - Dec - 2019].

[4] M. Asad,” Optimized Stock market prediction using ensemble learning”, IEEE dated 30 November 2019.

[5] B. Weng, L. Lu, X. Wang, F. M. Megahed, W. Martinez,” Predicting Short-Term Stock Prices using Ensemble Methods and Online Data Sources”, 2020.06.016.

[6]K. S. Zarkias ; N. Passalis ; A. Tsantekidis ; A. Tefas “Deep Reinforcement Learning for Financial Trading Using Price Trailing” IEEE dated 17 April 2020.

[5] S. M. Idrees ; M. A. Alam ; P. Agarwal “A Prediction Approach for Stock Market Volatility Based on Time Series Data” IEEE dated 25 January 2020.

[6] M. S. Hegde ; G. Krishna ; R. Srinath “An Ensemble Stock Predictor and Recommender System” IEEE dated 03 December 2021.

[7] X. Jiang ; S. Pan ; J. Jiang ; G. Long “Cross-Domain Deep Learning Approach For Multiple Financial Market Prediction” IEEE dated 15 October 2021.

[8] K. A. Althelaya ; El-Sayed M. El-Alfy ; S. Mohammed “Evaluation of bidirectional LSTM for short-and long-term stock market prediction” IEEE dated 07 May 2021.

[9] T. Ye “Stock forecasting method based on wavelet analysis and ARIMA-SVR model” 19 June 2020.

[10] C. A. S. Assis; A. C. M. Pereira; E. G. Carrano; R. Ramos; W. Dias” Restricted Boltzmann Machines for the Prediction of Trends in Financial Time Series” 978-1-5090-6014-6/18/$31.00 ©2018 IEEE.

[11] W. Bao; Xiao-Yang Liu,” Multi-Agent Deep Reinforcement Learning for Liquidation Strategy Analysis”, arXiv:1906.11046v1[q-fin.TR] 24 Jun 2019.

[12] C. Olah, “Understanding LSTM Networks”, August 27, 2017.

[13] M. Hessel; J. Modayil; H. Hasselt; T. Schaul; G. Ostrovski; W. Dabney; D. Horgan; B. lPiot; M. Azar; D. Silver;” Rainbow: Combining Improvements in Deep Reinforcement Learning”, [cs.AI] 6 Oct 2017.

[14] K. D. Asis, J. F. Hernandez-Garcia, G. Z. Holland, R. S. Sutton, “Multi-Step Reinforcement Learning: A Unifying Algorithm”, 11 Jun 2018.