CHAPTER 1

INTRODUCTION

Introduction

Forecasting the future has always been an difficult and facinating task for the probing individuals. This kind of forecasting becomes more fascinating when it involves money and risk like predicting the Stock Market. On the basis of the literature review, different attributes are selected for the prediction model. Most of the research work done in stock market is by the people related to business field. Although some research has also been made by computer science researchers but it still shows a vast area to be explored. Talking specifically about the National Stock Exchange, there are some amount of researchers who have used the machine learning and deep learning techniques for National Stock Exchange market prediction. Research has been done on Stock market forecasting by researchers of different fields including the business and computer science. Researchers tried various approaches for market prediction including different methods & algorithm and various combination of attributes. The attribute that makes a prediction model depends upon the factors on which market performance can depend. The main objective of this project is to find out whether the combination of different techniques that includes statistical, analytical and data mining techniques can predict stock market or not and up to what accuracy level.

1.1 Overview

This project will investigate how different Deep learning and machine learning methods can be used and will affect the accuracy of stock price forecasting. Various models, from regression to dense and recurrent neural netwoks are tested. Various hyperparameters are also tuned for better performance. The search space for all neural network architectures and hyperparameter combinations is vast, and with limited time to complete this project, the team optimises the models with evolution algorithms, replicating AutoML techniques from other researches with promising results in the financial context.

1.2 Motivation

Stock price prediction is a well-known and significant problem. With a successful stock prediction model, we can gain insight into market behaviour over time, spotting trends that would otherwise go unnoticed. Machine learning will be an efficient method to solve this problem with the increasing computational power of computers. However, the public stock dataset is too small for many machine learning algorithms to work with, and requesting more features could cost thousands of dollars per day. In this project, we will present a framework for incorporating user predictions into existing machine learning and deep learning algorithms while using public historical data to improve our results. The motivated idea is that if we know all of today's stock trading information (of all specific traders), the price is predictable. As a result, even if we only have partial information, we can expect to improve the current prediction lot. With the rise of the Internet, social networks, and online social interactions, predicting daily user behavior has become a viable job. As a result, our motivation is to create a public service that incorporates historical data and user predictions to create a stronger model that benefits everyone.

1.3 Objective

- The goal of Stock Market Prediction is to forecast the future value of a company's financial stocks.
    - The use of machine learning, which makes predictions based on the values of current stock market indices.
    - Machine learning employs various models to make prediction easier and more accurate.
    - The project focuses on predicting stock values using Regression, LSTM-based Deep Learning, and Machine Learning algorithms.
    - Open, close, low, high, and volume are all factors to consider.

CHAPTER 2

LITERATURE SURVEY

2.1 STUDY OF RESEARCH PAPER

In the paper “Stock Market’s Price Movement Prediction With LSTM Neural Networks” [2], the authors David M. Q. Nelson, Adriano C. M. Pereira, Renato A. de Oliveira have proposed that, The stock market is a very complicated, chaotic, and dynamic environment, making price predictions quite difficult. There are numerous studies from many of the areas attempting to tackle that difficulty have put their attention on machine learning techniques.

There are numerous instances where machine learning algorithms were successful in producing acceptable outcomes while making that kind of prediction. In order to forecast future stock price movements based on price history and technical analysis indicators, this article investigates the use of LSTM networks in that scenario. In order to achieve this, a prediction model was created, and a number of experiments were run. The results of these experiments were then compared to various metrics to determine whether this particular type of algorithm offers any advantages over other Machine Learning techniques and investment strategies. The results showed promise, reaching up to an average of 55.9% accuracy when forecasting whether the price of a certain stock will increase or decrease in the near future.

Stock market predictions have been the subject of research for many years, but they have shown to be highly challenging due to their inherent complexity, dynamism, and chaos. The there are a huge number of variables and information sources taken into account, making the signal-to-noise ratio low. Because of this, projecting future stock market price behavior is quite difficult.

The feasibility of such a feat has been discussed in Science for many years, and it's noteworthy in the associated literature that the majority of prediction models fall short of offering accurate predictions in a broad sense.Nevertheless, a significant number of studies from other fields are attempting to tackle this difficulty and are proposing a wide range of strategies for doing so.

Utilizing machine learning algorithms to learn from historical pricing data and forecast future prices is a typical strategy.This article takes a step in that direction by examining a particular approach that makes use of recurrent neural networks. Since these networks have short-term memory capabilities, it is hypothesized that they will perform better than other, more conventional approaches in the field of machine learning.

The Long-Short Term Memory (LSTM) network is the preferred algorithm in this situation. Given its capacity to distinguish between recent and early samples by giving different weights for each while forgetting memory it considers unimportant to forecast the next output, this sort of recurrent network has demonstrated excellent performance on a number of challenges.

Advantage: When observing the maximum losses it is possible to conclude that the LSTM based model offers less risks when compared to the other strategies.

Disadvantage: When compared to the other machine learning models it has displayed considerable gains in terms of accuracy, but in another hand we believe that variance could be lower and that would contribute for more reliable model.

In the paper “Predicting Stock Market Trends Using Machine Learning and Deep Learning Algorithms Via Continuous and Binary Data; a Comparative Analysis” [3] the authors Mojtaba Nabipour , Pooyan Nayyeri , Hamed Jabani have proposed that, For investors, the nature of stock market movement has always been unclear due to numerous deciding factors. With machine learning and deep learning algorithms, this study seeks to drastically lower the risk of trend prediction. the four stock market sectors,

For experimental evaluations, Tehran Stock Exchange's varied financials, petroleum, non-metallic minerals, and basic metals are chosen. Nine machine learning models (Decision Tree, Random Forest, Adaptive Boosting, eXtreme Gradient Boosting, Support Vector Classifier, Naive Bayes, KNearest Neighbors, Logistic Regression, and Artificial Neural Network (ANN)) and two potent deep learning techniques (Recurrent Neural Network (RNN) and Long Short-Term Memory) are compared in this study (LSTM). Our input values come from ten technical indicators extracted from ten years of historical data, and there are two methods for using them. First, stock trading values are used to calculate the indicators, which are then converted to binary data before use. Based on the input methods, three metrics are used to assess each prediction model. The evaluation findings show that for continuous data, RNN and LSTM perform significantly better than other prediction models. Results also indicate that those deep learning techniques are the best for evaluating binary data; nevertheless, the difference between them is diminishing as a result of the second method's clearly improved model performance.

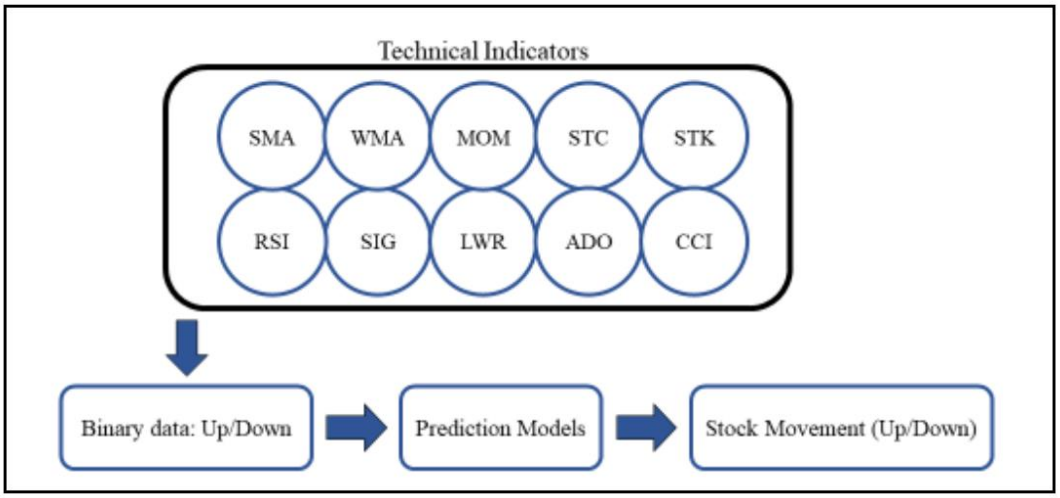


Figure 2.1 : Predicting stock movement with binary data.

SMA indicators are used by investors to determine if a price will continue to move in the same direction by computing the average of values inside a specific range.

SMA and WMA: the trend is -1 if the current value is lower than the moving average, and +1 otherwise.

There are typically two approaches to stock market forecasting. One of them is fundamental analysis, which depends on a company's strategy and fundamental data

including market position, costs, and annual growth rates. The second method focuses on historical stock prices and values and is known as technical analysis.

It is obvious that unforeseen factors, such as a company's reputation or a nation's political climate, can alter the direction of the stock market. Therefore, the trend of stock values and the index can be anticipated if the data obtained from stock values is effectively preprocessed and appropriate algorithms are used. Machine learning and deep learning techniques can aid traders and investors in making decisions in stock market prediction systems.

According to the final findings, SVM ensemble performed significantly better for classification than solo SVM. The value trends of the Hang index from the Hong Kong market were predicted by Out using ten data mining techniques. Bayesian classification, SVM, Knearest neighbor, neural networks, and tree-based classification were the techniques used. Results showed the SVM performed better than other predictive models.

Pang used LSTM with an integrated layer and an automated encoder to obtain more accurate stock market estimates. The results of experimental investigation showed that LSTM with an embedded layer outperformed for the Shanghai composite index with 57.2% accuracy.

Advantage: As a prominent result, deep learning methods (RNN and LSTM) show a technical skill to forecast stock movement in both approaches, especially for continuous data. Experimental works showed that there was a significant improvement in the performance of models when they use binary data instead of continuous one.

Disadvantage: It can be seen that Naive-Bayes and Decision Tree are least accurate so machine learning models are not more consider for binary data over continuous data. While recurrent neural network and long short term memory model are top predictors which are deep learning models in binary data.

In the paper “Stock Price Prediction using Machine Learning and Sentiment Analysis” [4] the authors Yash Mehta, Atharva Malhar, Dr. Radha Shankarmani have proposed that, Predicting the future value of a company's financial stocks is the goal of stock market prediction. The use of machine learning in stock market prediction technologies is a recent development. This technology produces forecasts based on the values of current stock market indices by training on their past values. To facilitate and authenticate prediction, machine learning itself uses a variety of models. The main topic of the research is how to estimate stock values using machine learning that is LSTM and regression based. Open, close, low, high, and volume are all taken into consideration.

For the seller and the broker, a successful stock forecast might result in enormous gains. The argument that prediction is chaotic rather than random is frequently made, and as a result, it is possible to make predictions by carefully reviewing the history of the relevant stock market. The most effective technique to depict such processes is through machine learning. The accuracy is increased by the prediction of a market value that is near to the tangible worth. The application of machine learning to stock prediction has attracted the attention of numerous researchers due to its effective and precise assessments.

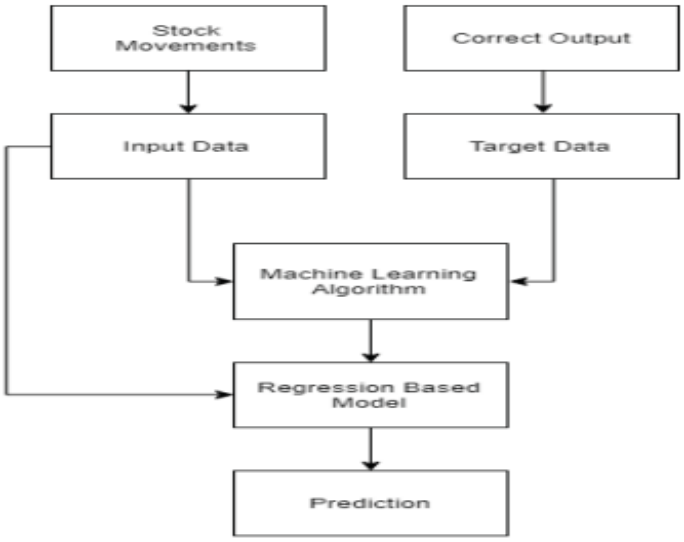


Figure 2.2: Flow Chart for Regression Based Model

The dataset used in machine learning is crucial. Since a small change in the data might lead to significant changes in the results, the dataset should be as precise as possible.

On a dataset collected from Yahoo Finance, this study uses supervised machine learning. The open, close, low, high, and volume variables make up this dataset. The bid prices for the stock at the open, close, low, and high are all different and have nearly identical names. The volume represents the total number of shares that changed hands within the time frame. After that, the model is evaluated using the test data.

For this hypothesis, independent use of the LSTM and regression models is made. Regression includes reducing error, and LSTM helps by retaining the data and outcomes over time. Finally, graphs are shown showing the relationship between

actual and anticipated prices (for the LSTM-based model) and the variation of prices with dates (in the case of a regression-based model).

Advantage: The results clearly show that the ARIMA model is having the least accuracy. The ARIMA model gives best results when forecasting short term results whereas the LSTM model is better for long term predicating.

Disadvantage: For the stock market prediction we require short term prediction to be more accurate as the user has to run the model daily for each stock, so for this particular scenario we do not require long term prediction to be more accurate and hence the ARIMA model is giving the best RMSE score.

In the paper “Market Prediction Using Machine Learning” [5] the authors Ishita Parmar, avanshu Agarwal, Sheirsh Saxena, Ridam Arora, Shikhin Gupta, Himanshu Dhiman, Lokesh Chouhan have proposed that, Predicting the future value of a company's financial stocks is the goal of stock market prediction. The use of machine learning in stock market prediction technologies is a recent development. This technology produces forecasts based on the values of current stock market indices by training on their past values. To facilitate and authenticate prediction, machine learning itself uses a variety of models. The main topic of the research is how to estimate stock values using machine learning that is LSTM and regression based. Open, close, low, high, and volume are all taken into consideration.

The dataset used in machine learning is crucial. Since a small change in the data might lead to significant changes in the results, the dataset should be as precise as possible.

On a dataset collected from Yahoo Finance, this study uses supervised machine learning. The open, close, low, high, and volume variables make up this dataset. The bid prices for the stock at the open, close, low, and high are all different and have nearly identical names. The volume represents the total number of shares that changed hands within the time frame. After that, the model is evaluated using the test data.

For this hypothesis, independent use of the LSTM and regression models is made. Regression includes reducing error, and LSTM helps by retaining the data and outcomes over time. Finally, graphs are shown showing the relationship between actual and anticipated prices (for the LSTM-based model) and the variation of prices with dates (in the case of a regression-based model). The following is the remainder of the paper: The relevant work is covered in Section II. The two models and their respective techniques are presented in detail in Section III. The results produced with various plots for both models are thoroughly covered in Section IV. While Section V is the conclusion and Section VI is where the references are found.

Advantage: The LSTM Model offered more accuracy than the Regression based Model. This paper was an attempt to determine the future prices of the stocks of a company with greater accuracy and reliability using machine learning techniques. The primary contribution of the researchers being the application of the novel LSTM Model as a means of determining the stock prices.

Disadvantage: Since stock market involves processing of huge data, the gradients with respect to the weight matrix may become very small and may degrade the learning rate. This corresponds to the problem of Vanishing Gradient.

In the paper “Stock Market Analysis using Supervised Machine Learning” [6] the authors Kunal Pahwa, Neha Agarwal have proposed that, One of the most complex and sophisticated methods of conducting business is the stock market or share market. A very complex model, small ownerships, brokerage firms, and the banking industry all rely on this one body to generate money and distribute risks. However, this research suggests employing machine learning algorithms to forecast future stock prices for exchange by utilizing open source libraries and current algorithms to help make this erratic business model a little more predictable. We'll see if the results of this straightforward implementation are satisfactory. The result is entirely determined by math, and it makes numerous assumptions about the real world that may or may not hold true at the time of prediction.

One of the first ways for a regular individual to trade stocks, make investments, and profit from businesses that sell a piece of themselves on this platform is the STOCK MARKET. If used carefully, this strategy demonstrates its potential as an investment scheme.

Figure 2.3 : Googles Accuracy vs Price

Graph showing stock price of GOOGLE from year 2005 till July 2018. Red is the line representing given data and blue is representing the forecasted or the predicted value of stock.

Although this platform's prices and liquidity are highly unpredictable, we use technology to our advantage in this situation. One such tool that assists us in achieving our goals is machine learning.

As everyone is aware, the stock market is a crucial trading venue that has an impact on everyone both locally and nationally. The fundamental idea is really straightforward: Businesses will list their shares as tiny commodities called Stocks on the stock exchange. To raise money for the business, they act in this manner. At a price known as the IPO, or initial public offering, a company lists its stock. At this offer price, the corporation will sell its stock and raise capital.

After then, the owner owns the stock and is free to sell it to a buyer at any price on a stock exchange like the Bombay Stock Exchange or BSE. Traders and buyers continue selling these shares at their own price but the company only gets to keep the money made during the IPO. The continue hoping of hares from one party to another in order to make more profits, results in an increase of price of the particular share after every profitable transaction. However, if the company issues more stocks at a lower IPO, then the market price for exchange goes down and traders suffer a loss. This exact phenomenon is the reason for the fear people have in investing in stock markets and the reason for the fall and rise of stock prices in a nutshell.

In statistics, we can determine the dependent and independent variables and attempt to establish or recognize an existing link between them by examining the values and characteristics of a problem represented as a graph. Due to its extremely straightforward and efficient method, this technique is known as linear regression in statistics and is widely utilized. In machine learning, we have modified the same process where we utilize the features to train the classifier which then predicts the value of the label with a given accuracy which can be checked when training and testing of the classifier.

You must choose the proper features and have enough data to train your classifier in order for it to be accurate. Your classifier's accuracy directly relates to the attributes you choose and the amount of data you feed it.

Advantage: Shifting to SVM, trying and testing different models, looking for new and improved features, changing the entire data model to suit the model entirely etc. are some very fundamental ways to optimize your classifier.

Disadvantage:

- Make sure your test data is at least 20% of the size of your training data. It is important to understand that testing is the test of you classifiers accuracy and is sometimes observed to be inversely proportional to a classifiers score.
- This paper is limited to only supervised machine learning, and tries to explain only the fundamentals of this complex process.

In the paper “Stock Market Prediction Using Machine Learning Techniques” [7] the authors Mehak Usmani , Syed Hasan Adil , Kamran Raza, Syed Saad Azhar have proposed that, The main objective of this research is to predict the market performance of Karachi Stock Exchange (KSE) on day closing using different machine learning techniques. The prediction model uses different attributes as an input and predicts market as Positive & Negative. The attributes used in the model includes Oil rates, Gold & Silver rates, Interest rate, Foreign Exchange (FEX) rate, NEWS and social media feed. The old statistical techniques including Simple Moving Average (SMA) and Autoregressive Integrated Moving Average (ARIMA) are also used as input. The machine learning techniques including Single Layer Perceptron (SLP), Multi-Layer Perceptron (MLP), Radial Basis Function (RBF) and Support Vector Machine (SVM) are compared. All these attributes are studied separately also. The algorithm MLP performed best as compared to other techniques. The oil rate attribute was found to be most relevant to market performance. The results suggest that performance of KSE-100 index can be predicted with machine learning techniques.

Performance can depend. On the basis of the literature review, different attributes are selected for the prediction model. Most of the research work done in stock market is by the people related to business field. Although some research has also been made by computer science researchers but it still shows a vast area to be explored.

In Computer Science the prediction may often relates to data mining or machine learning. Talking specifically about the Karachi Stock Exchange, there are very few researchers who have used the machine learning techniques for KSE market prediction. The main objective of this research is to find out whether the combination of different techniques that includes statistical, analytical and data mining techniques can predict stock market or not and up to what accuracy level.

Advantage: The Multi-Layer Perceptron algorithm of machine learning predicted 77% correct market performance. Even with the lack of resources and unavailability of data for the market, the model was able to predict the performance of the model to a good extent with only 100 instances that shows that karachi stock exchange has the tendency to be predicted using machine learning techniques.

Disadvantage: Single Layer Perceptron model trained by training set and tested on the different data set. The model gave about 60% accurate results. The model was unable to predict the Positive class well. When the SLP algorithm was tested on same data set that was used for training, it gave 83% accuracy.

In the paper “Stock Trend Prediction Using Candlestick Charting and Ensemble Machine Learning Techniques With a Novelty Feature Engineering Scheme” [8] the authors Yaohu Lin , Shancun Liu , Haijun Yang ,Harris Wu. Have proposed that, Due to the extremely noisy, nonparametric, complicated, and chaotic character of stock price time series, predicting the stock market is a difficult endeavour. We build a unique ensemble machine learning framework for daily stock pattern prediction using a basic eight-trigram feature engineering strategy of inter-day candlestick patterns, merging classical candlestick charting with the most recent artificial intelligence technologies. To anticipate the direction of the closing price, several machine learning approaches, including deep learning methods, are used to stock data. Based on the training findings, this framework may recommend a machine learning prediction approach for each pattern. The investing plan is built using ensemble machine learning techniques. The empirical outcomes of China's stock market from 2000 to 2017 demonstrate that our feature engineering is effective. Nonetheless, transaction costs have a significant impact on investment. Additional technical indicators can improve the forecast accuracy to varying degrees. Technical indicators, especially momentum indicators, can improve forecasting accuracy in most cases.

Forecasting the stock market is an essential goal in the financial world, and it remains one of the most difficult challenges owing to the non-linear and chaotic structure of the financial sector. Stock market investments are frequently led by various prediction methods, which may be separated into two categories: technical analysis and fundamental analysis. The basic analysis technique is concerned with the company, and it makes use of the firm's economic position, workers, annual reports, financial status, balance sheets, income reports, and so on. Technical analysis, also known as charting, on the other hand, forecasts the future by analysing trends in previous data. Using technical analysis tools, investors might develop effective trading strategies. Candlestick charting, which uses open-high-low-close prices, can represent not just the.

Advantage:

- SVM, RF and GBDT showed good predictions for all patterns of technical chart
- The increase in the number of parameters can increase the level of linear prediction, which also reflects the complexity and diversity of financial markets from another perspective.
- the momentum indicators are significantly better via empirical testing than other indicators in short-term forecasting

Disadvantage:

- The forecasting framework of this paper has predictive power, although it is difficult to profit from certain patterns due to the stop-trading rules of the Chinese market.

In the paper “Which Artificial Intelligence Algorithm Better Predicts the Chinese Stock Market?” [9] the authors Lin Che, Zhilin Qio , Mingang Wang , Chao wong have proposed that, Unpredictable stock market dynamics make stock index futures difficult to forecast. Although efforts to design an effective prediction approach have a long history, current advances in artificial intelligence and the usage of artificial neural networks have boosted our effectiveness in nonlinear approximation. When studying financial markets, we may now extract characteristics from a large data environment without previous predictive knowledge. We propose here to improve on this predictive performance by combining a deep-learning-based stock index futures prediction model, an autoencoder, and a limited Boltzmann machine. We assess three standard artificial neural networks' predicting performance using high-frequency data: Unpredictable stock market dynamics make stock index futures difficult to forecast. Although efforts to design an effective prediction approach have a long history, current advances in artificial intelligence and the usage of artificial neural networks have boosted our effectiveness in nonlinear approximation. When studying financial markets, we may now extract characteristics from a large data environment without previous predictive knowledge. We propose here to improve on this predictive performance by combining a deep-learning-based stock index futures prediction model, an autoencoder, and a limited Boltzmann machine. We assess three standard artificial neural networks' predicting performance using high-frequency data:

Stock market forecasting is a well-known issue in both business and academic circles. Extreme stock market movements, such as the February 2018 global stock market upheaval, harm financial markets and the global economy. As a result, we require a more accurate method of forecasting market swings. Among the many predictive efforts over the last few decades, some have had success using quantitative methods such as autoregressive integrated moving average (ARIMA) models, artificial neural networks, support vector machines, and neuro-fuzzy based systems, but financial economists continue to debate these methodologies due to the nonlinear characteristics of stock market behaviour.

"Deep learning" (DL) has recently received a lot of interest in various scientific domains. DL is a novel branch of machine learning that has enhanced computers' abilities in image identification and classification, natural language processing, audio recognition, and social network filtering. In certain circumstances, the findings are equivalent to, if not better than, those of human specialists. The structure of DL is a multi-layer neural network that extracts and transforms diverse characteristics using a cascade of many layers of nonlinear processing units. It can learn either supervised or unsupervised, and it creates a concept hierarchy by employing numerous representation layers that correspond to different degrees of abstraction. The usage of DL has increased processing power and huge data storage Without adopting econometric assumptions, DL may extract abstract characteristics and find hidden linkages in financial markets. Traditional financial economic procedures and other quantitative techniques are incapable of doing this. in opposition to limitations of existing models Deep learning can process high frequency big data, analyze the financial market, and predict stock returns. To get optimal outcomes, the learning rate, epochs, targets, and amount of artificial neurons must all be considered while defining the parameters of artificial neural networks. Without subjective interferences, DL can extract abstract characteristics. When we employ a huge time-series dataset to anticipate financial market movement, we don't must include influencing elements or

control variables. Other approaches for time-series analysis have been used. For example, proposes a unique wavelet multiresolution complex network for evaluating multivariate time series, which is capable of analyzing information in both the dynamical and topological sectors and has been effectively employed in a variety of study disciplines. Nonetheless, DL can deliver reliable financial market time-series forecasting, and the accuracy rises as the database size grows.

Advantage:

- The extreme learning machine (ELM) is used for classification, regression, clustering, and feature learning with a single layer or multi layers of hidden nodes in which the parameters of hidden nodes need not be tuned.
- The predictive accuracy of DL increases as the sample volume increases. Although the DL running time consumed is more than that of BP and ELM, it is much less than that of RBF, and although RBF is more accurate, it requires so much more running time and this advantage is obviated

Disadvantage:

- Despite these advantages, ELM cannot be used when the time series is noisy.

CHAPTER 3

PROBLEM STATEMENT

3.1 Problem Statement

To predict the stock market ups and downs using Machine Learning (Linear regression, Random forest) and Deep Learning Algorithms (LSTM, GRU).

CHAPTER 4

PROJECT REQUIREMENT

4.1 System Requirement

4.1.1 Hardware Requirement:

RAM: 8 GB

As we are using Machine Learning Algorithm and Various High-Level Libraries

LAPTOP: RAM minimum required is 8 GB.

PROCESSOR: Intel i5 Processor or AMD Ryzen 5 Processor

Jupyter Notebook and **Google Collab** is to be used and data loading should be fast hence Fast Processor is required

CODING LANGUAGE: Python Version 3.5 at least

Highly specified Programming Language for Machine Learning and Deep Learning because of availability of High-Performance Libraries.

OPERATING SYSTEM: Windows 10 at least

Latest Operating System that supports all type of installation and development Environment

4.1.2 Software Requirement

Operating System: Windows 10

IDE: Jupyter Notebook and Google Collab

Best Integrated Development Environment as it gives possible suggestions at the time of typing code snippets that makes typing feasible and fast.

Programming Language: Python

Highly specified Programming Language for Machine Learning and Deep Learning because of availability of High-Performance Libraries

4.2 FUNCTIONAL REQUIREMENT

4.2.1 System Feature 1(Functional Requirement)

Functional requirement describes features, functioning, and usage of a product/system or software from the perspective of the product and its user. Functional requirements are also called as functional specifications where synonym for specification is design. Provide User friendly Interface and Interactive as per standards.

1. Dataset: Training deep learning models requires extensive data samples for the model to achieve high accuracy and efficiency. The project uses Echocardiogram images Dataset

2. Data Pre-processing capability: Training Deep Learning models on large datasets requires effective data pre-processing of the dataset under consideration. We have use various data pre-processing and data augmentation techniques for the same.

3. Echocardiogram Detection capability: After pre-processing, a Deep Learning model is trained for effective recognition of Echocardiogram.

4. Model Interpretability: For the lung cancer recognition system to be interpretable, various deep learning algorithms are used to explain the inner workings of the ML model.

5. Web interface: For the end user to use this Echocardiogram detection system, a web interface is provided to access and use the system

6. High Speed: System should process requested task in parallel for various action to give quick response. Then system must wait for process completion.

7. Accuracy: System should correctly execute process, display the result accurately. System output should be in user required format.

4.3 NON-FUNCTIONAL REQUIREMENT

4.3.1 Performance Requirements

The performance of the functions and every module must be well. The overall performance of the software will enable the users to work evidently. Performance of encryption of data should be fast. Performance of the providing virtual environment should be fast Safety Requirement. The application is designed in modules where errors can be detected and easily. This makes it easier to install and update new functionality if required.

4.3.2 Safety Requirements:

The application is designed in modules where errors can be detected and fixed easily. This makes it easier to install and update new functionality if required. The data safety must be ensured by arranging for a secure and reliable transmission media. The source and destination information must be entered correctly to avoid any misuse or malfunctioning. Password generated by user is consisting of characters, special character number so that password is difficult to hack. So, that user account is safe.

4.3.3 Security Requirements

Secure access of confidential data (user's details). Information security means protecting information and information systems from unauthorized access, use, disclosure, disruption, modification, or destruction. The terms information security, computer security and information assurance are frequently incorrectly used interchange- ably. These fields are interrelated often and share the common goals of protecting the confidentiality, integrity, and availability of information; however, there are some subtle differences between them.

4.3.4 Software Quality Attributes \

1. Runtime System Qualities: Runtime System Qualities can be measured as the system executes.

2. Functionality: The ability of the system to do the work for which it was in- tended.

3. Performance: The response time, utilization, and throughput behaviour of the system. Not to be confused with human performance or system delivery time.

4. Security: A measure of system's ability to resist unauthorized attempts at usage or behaviour modification, while still providing service to legitimate users.

5. Availability: (Reliability quality attributes fall under this category) the measure of time that the system is up and running correctly; the length of time between failures and the length of time needed to resume operation after a failure.

6. Usability: The ease of use and of training the end users of the system. Sub qualities: learn ability, efficiency, affect, helpfulness, control.

7. Interoperability: The ability of two or more systems to cooperate at runtime.

CHAPTER 5

SYSTEM ANALYSIS

5.1 System Architecture

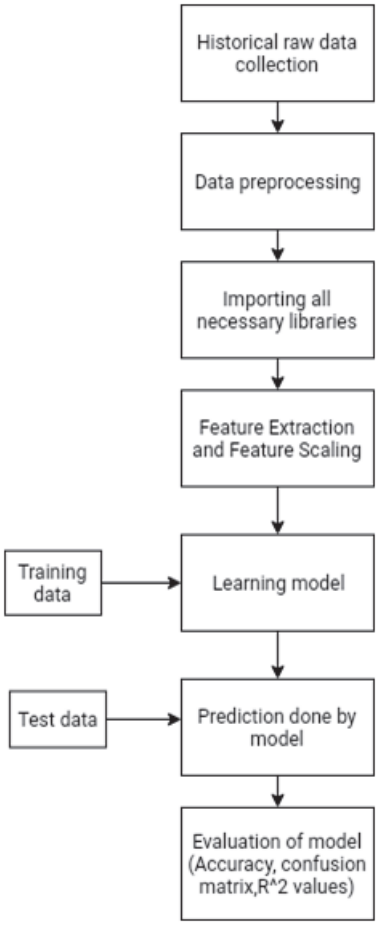


Figure 5.1 : Flowchart of application

1. The raw data used is from Yahoo finance [10].

2. The attributes used for feature extraction are ‘date’ and ‘closing price’ of a stock.

3. Features used to predict the momentum of stock price of particular company are ‘stock momentum’, ‘index volatility’, ‘sector momentum’. These features are scaled.

4. The dataset is split into training and test data set.

5. The training dataset is used for model training and test dataset is used for prediction .The significance of a feature is determined using the R^2 values

6. The values of the test data are predicted and the results are evaluated. The result is given on the basis of accuracy, confusion matrix and time required for the model used

1. **Import libraries**

The following libraries must be installed before getting started.

1. **Pandas** — a python library that loads the data file as a pandas data frame for analyzing the data.
2. **Matplotlib**— a python package for plotting graphs.
3. **Scikit-learn —** an open-source python library used in data analysis that supports machine learning models, pre-processing, model evaluation, and training utilities. It also acts as a sub-library for train_test_split, RandomForestRegressor, StandardScaler, RandomizedSearchCV, and metrics.
4. **Numpy**— a python library that works with arrays.
5. **Yfinance** — a python open-source library used to access financial data.

2. Import the dataset

The information needed for this project is the historical data. It contains the date, open price, close price, highest price, lowest price, and volume of the trades for each trading day. Traders use these data to measure the volatility of a stock.

Features	Meaning
Date	The stock value date
Open	Open price of the stock, at the beginning of trading day
High	Highest point of the stock price, on a trading day.
Low	Lowest point of the stock price, on a trading day.
Close	Close price of the stock, at the end of a trading day.
Adj close	Amended closing price for dividends of stock value the stock's value after distributing dividends.
Volume	Number of traded stocks in the market over a period

The data is got via a python script. Here, a python script is used to gain the data by using yfinance. It will fetch the Google stock data from 1st January 2019 to 22th February 2023. The downloaded Google stock data is loaded into a data frame and converted into a CSV file (comma separate value). So, that I can store it locally and quickly load it into the algorithm.

3. Visualize the data

Plot a line chart of the adjusted close prices over time.

Result :

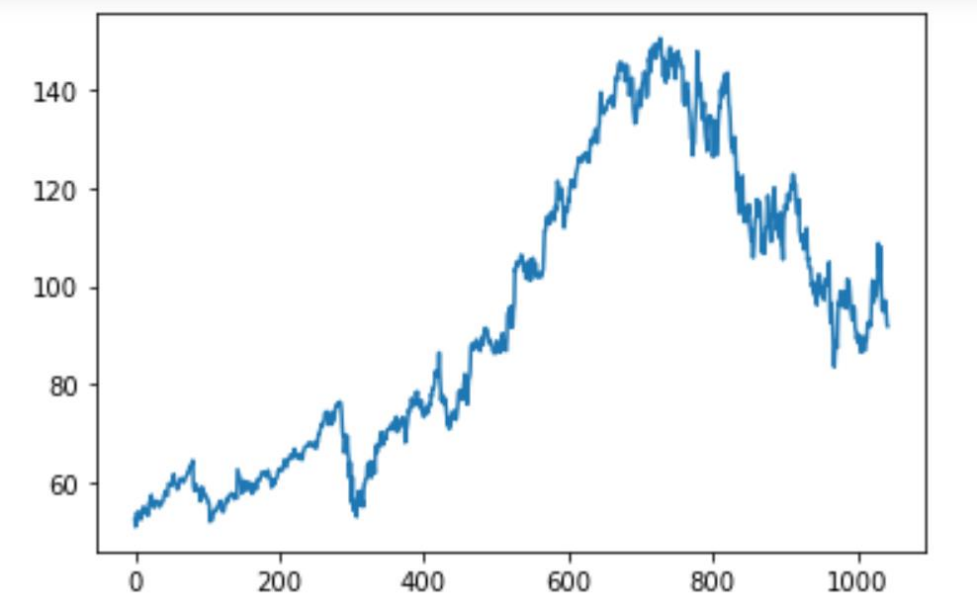


Fig 5.2 Graph of Google stock data

4. Data pre-processing

This step is the most important part of this project. Data pre-processing are steps taken to make the data ready for the machine learning model. Pre-processing includes transforming the raw data into a format that the model can take from and operate on. This project aims to have a dataset that the model accepts, and the algorithm understands. In a dataset, values can be missing, and information can be redundant and irrelevant, or noisy. Data cleaning is a form of pre-processing that includes removing missing or inconsistent values and changing the index. As well as feature selection, hyperparameter tuning, and data standardization

.

The model needs to work with what you are giving it.

Step 1: Read the file and set the date as the index.

Step 2: Feature selection

This is where the x and y features are selected to achieve the data set for the model. x and y features are declared for the training and testing data set.

Features are columns in the dataset. Feature selection is one of the essential concepts of machine learning applications that have a tremendous impact on the model's performance. In future selection, not every column is necessary. These features selected have an impact and contribute to the prediction output. Unnecessary features decrease the general performance of the test **set.** A method of selecting futures is finding out the most important features, feature importance. Sklearn has feature importance and feature selector module that can be implemented. With the feature importance module, a score is given to each feature in the data. Features with the highest scores are the most relevant, and sound output variables are guaranteed. The benefits of using feature selection include improving accuracy, reducing overfitting, reducing training time, and

improving data visualization. The more features there are, the model is likely to suffer from overfitting.
x is holding values for the open, high, low, close and adj close columns. y is holding values for the adj-close column. The other columns, including that of volume, were not selected for the process because they will not be needed.

Step 3: Divide into train and test datasets.

The dataset must be split into a training and testing dataset before modeling.

Training set: is a subset of the dataset used to build and fit predictive models. A training set is generated by building a training dataset script, which generates the training set features from the input options and the raw stock price data. The data is fed into the model for training. The model learns from this data and runs on the train set.

Testing set: is a subset of the dataset to assess the likely future performance of a model. It is a good standard for evaluating the model. Testing set is used against the predicted dataset and testing the model that is trained. The model has not seen this portion of the set. It is used for evaluation purposes.

Step 4: Scaling the features

This is called data standardization. Sklearn has a function called standard scaler that is used for standardizing the dataset. Standardization is known to improve the numerical stability of the model and increase training speed.
”It is a good practice to fit the scaler on the training data and then transform the testing data. This would avoid any data leakage during the model testing process. This is useful when you want to compare data that correspond to different units.”
We are scaling the x_train and x_test using the standard scaler.

Step 5: Hyperparemater tuning

Hyperparameters are the settings of the model. It is important to have parameters that are adjusted for performance optimization. I set it after training and testing the dataset, before fitting and predicting. Hyperparameters solve the main problem in machine learning which is overfitting. I used random search cross-validation in this project.

The hyperparameters in the random forest model are either used to increase the predictive power of the model or to make the model faster.

For a random forest regression model, the best parameters to consider are:

1. **n_estimators** — number of trees in the forest
2. **max_depth** — maximum depth in a tree
3. **min_samples_split** — minimum number of data points before the sample is split

4. **min_samples_leaf** — minimum number of leaf nodes that are required to be sampled
5. **bootstrap** — sampling for data points, true or false
6. **random_state** — generated random numbers for the random forest.

5.2 Algorithms

In the forecasting of stock market project we decided to try prediction on Machine learning and deep learning algorithms. We choose here two machine learning and two deep learning algorithms. Machine learning models namely regression model and random forest regressor model whereas deep learning models are long short term model (LSTM) and gated recurrent unit (GRU) model. Let's see these model some basic information about these models in the development of project.

A) Linear Regression:

Linear Regression is an algorithm that belongs to supervised Machine Learning. It tries to apply relations that will predict the outcome of an event based on the independent variable data points. The relation is usually a straight line that best fits the different data points as close as possible. The output is of a continuous form, i.e., numerical value. For example, the output could be revenue or sales in currency, the number of products sold, etc. In the above example, the independent variable can be single or multiple.

Since the Linear Regression algorithm represents a linear relationship between a dependent (y) and one or more independent (y) variables, it is known as Linear Regression. This means it finds how the value of the dependent variable changes according to the change in the value of the independent variable. The relation between independent and dependent variables is a straight line with a slope.

Linear Regression Terminologies

1. **Cost Function**

The output which is obtained or predicted by an algorithm is referred to as yˆ (pronounced as yhat). The difference between the actual and predicted values is the error, i.e., y - yˆy^. Different values of y- yˆy^ (loss function) are obtained when the model repeatedly tries to find the best relation. The average summation of all loss function values is called the cost function. The machine learning algorithm tries to obtain the minimum value of the cost function. In other words, it tries to reach the global minimum.

$$minimize\frac{1}{n}\sum_{i=1}^{n}(pred_i - y_i)^2$$

$$J = \frac{1}{n}\sum_{i=1}^{n}(pred_i - y_i)^2$$

where J = cost function, n= number of observations (i = 1 to n), ∑ = summation, predi = predicted output and yi = actual value.

2. Gradient Descent

Another important concept in Linear Regression is Gradient Descent. It is a popular optimization approach employed in training machine learning models by reducing errors between actual and predicted outcomes. Optimization in machine learning is the task of minimizing the cost function parameterized by the model's parameters. The primary goal of gradient descent is to minimize the convex function by parameter iteration.

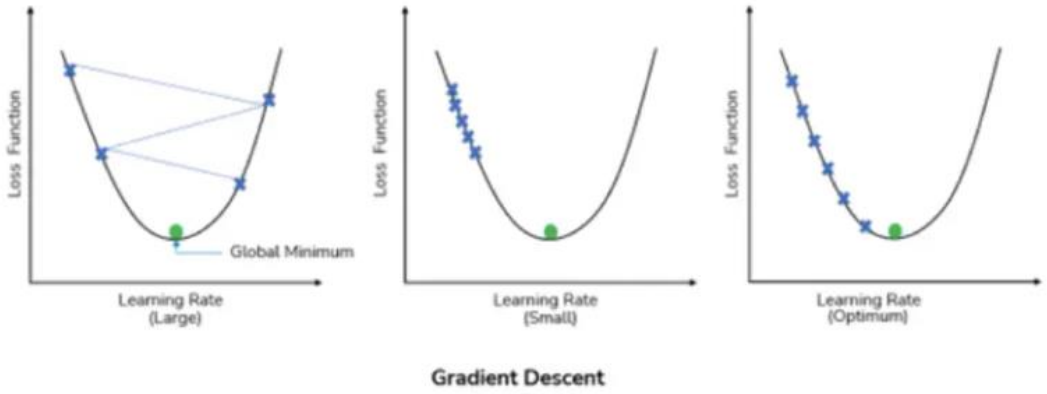


Fig. 5.3 Gradient Descent

A slower learning rate helps to reach the global minimum but takes an unusually long time and computationally proves expensive. The faster learning rate may make the model wander and lead to an undesired position, making it difficult to come back on the correct track to reach the global minimum. Hence the learning rate should be neither too slow nor too fast if the global minimum is to be reached

B) Random Forest

What is a Random Forest?

Random forest is a supervised classification machine learning algorithm which uses ensemble method. Simply put, a random forest is made up of numerous decision trees and helps to tackle the problem of overfitting in decision trees. These decision trees are randomly constructed by selecting random features from the given dataset.

Random forest arrives at a decision or prediction based on the maximum number of votes received from the decision trees. The outcome which is arrived at, for a

maximum number of times through the numerous decision trees is considered as the final outcome by the random forest.

Working of Random Forest

Random forests are based on ensemble learning techniques. Ensemble, simply means a group or a collection, which in this case, is a collection of decision trees, together called as random forest. The accuracy of ensemble models is better than the accuracy of individual models due to the fact that it compiles the results from the individual models and provides a final outcome.
How to select features from the dataset to construct decision trees for the Random Forest?

Features are selected randomly using a method known as bootstrap aggregating or bagging. From the set of features available in the dataset, a number of training subsets are created by choosing random features with replacement. What this means is that one feature may be repeated in different training subsets at the same time.

For example, if a dataset contains 20 features and subsets of 5 features are to be selected to construct different decision trees then these 5 features will be selected randomly and any feature can be a part of more than one subset. This ensures randomness, making the correlation between the trees less, thus overcoming the problem of overfitting.

Once the features are selected, the trees are constructed based on the best split. Each tree gives an output which is considered as a ‘vote’ from that tree to the given output. The output which receives the maximum ‘votes’ is chosen by the random forest as the final output/result or in case of continuous variables, the average of all the outputs is considered as the final output.

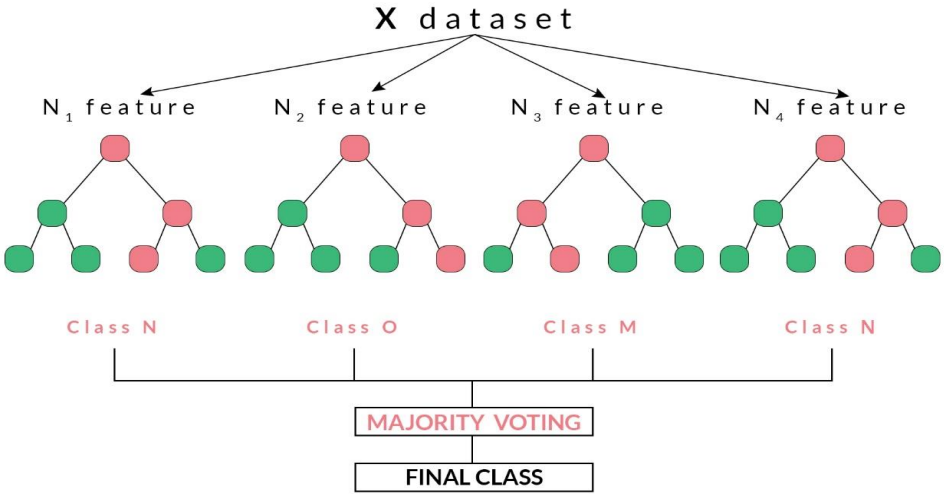


Fig 5.4 Random Forest

For example, in the above diagram, we can observe that each decision tree has voted or predicted a specific class. The final output or class selected by the Random Forest

will be the Class N, as it has majority votes or is the predicted output by two out of the four decision trees.

C) LSTM

Many papers demonstrate only the internal structure of LSTM — this may still leave you confused about how the system works. Thus I draw a complete diagram of LSTM in Figure (F.2), similar to that of RNN in Figure (D.2), to show you the whole system.

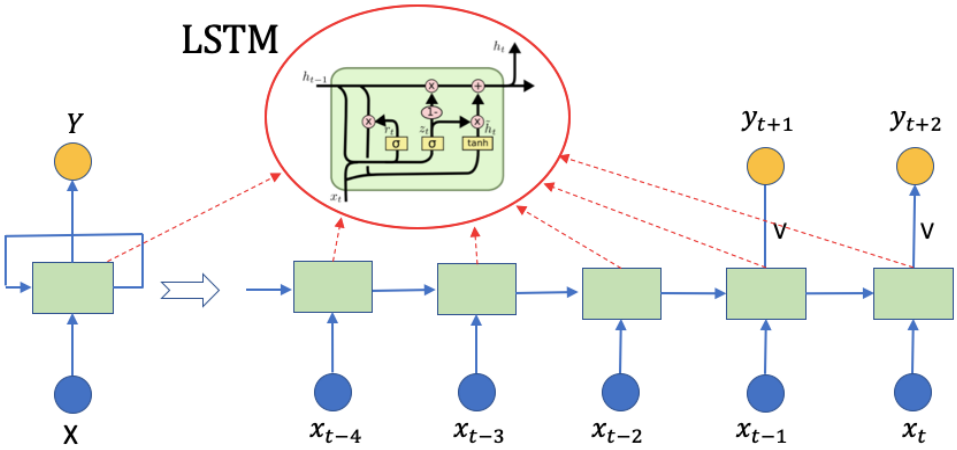
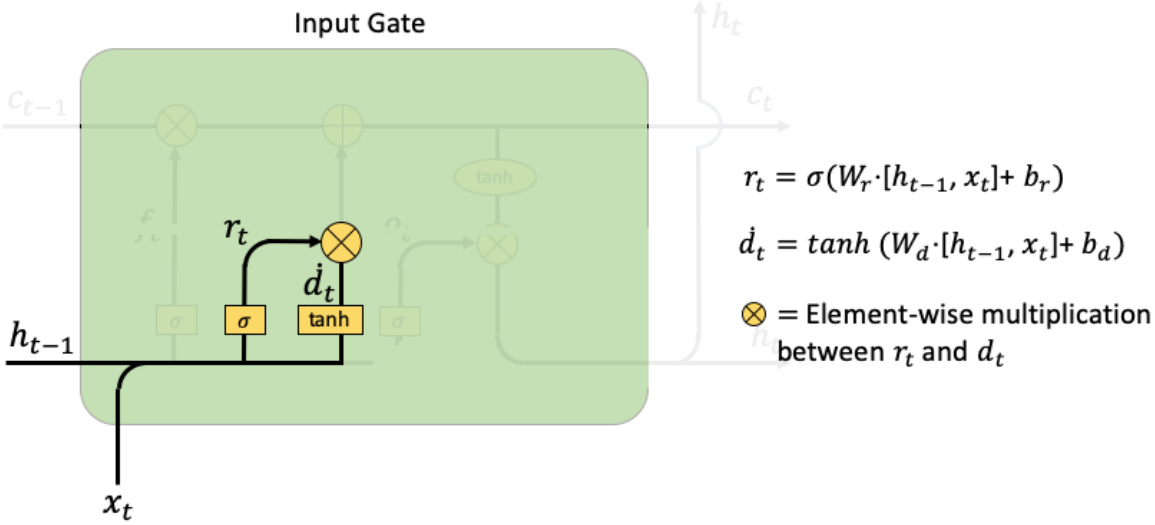


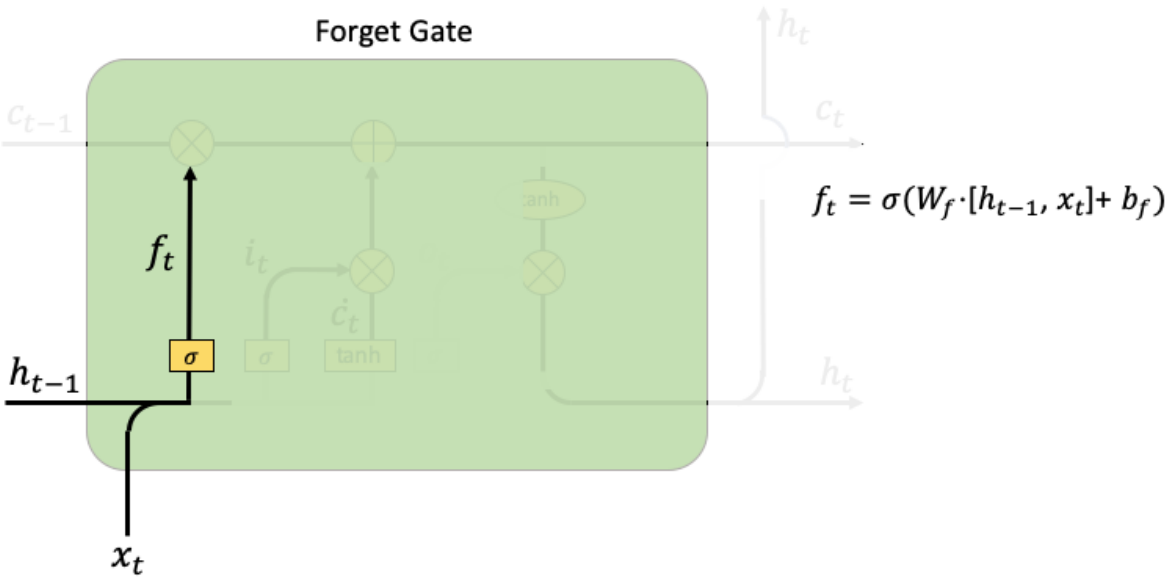
Fig. 5.5 LSTM Structure

The LSTM has four components: input gates, forget gates, cell state, and output gates.

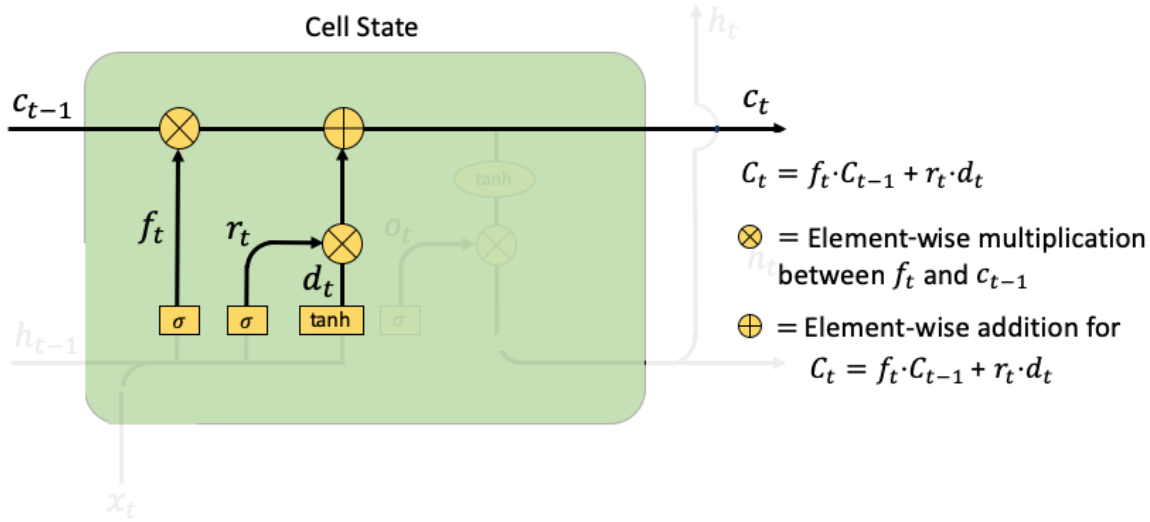
- Input Gate**:** the goal is to take in new information *xt*. There are two functions to take in new information: *rt* and *dt*. The *rt* concatenates the previous hidden vector *ht-1* with the new information *xt*., i.e., [*ht-1*, *xt*], then multiplies with the weight matrix *Wr*, plus a noise vector *br*. The dt does something similar. Then *rt* and *dt* are multiplied element-wise to get the cell state *ct*.



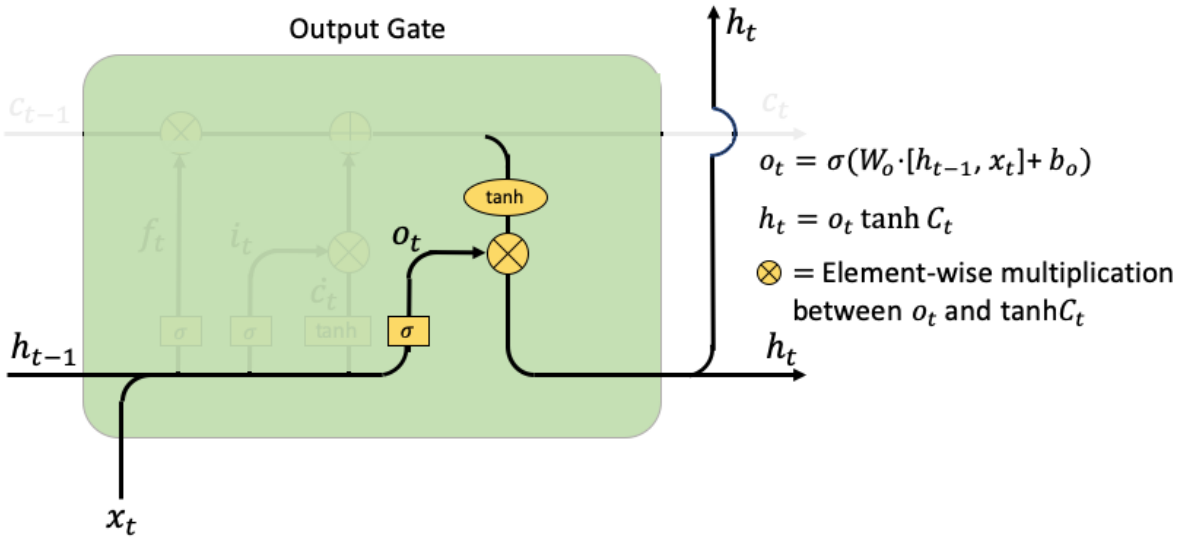
- Forget Gate: The forget gate *ft* looks very similar to *rt* in the input gate. It controls the limit up to which a value is retailed in the memory.



- Cell State: it calculates an element-wise multiplication between the previous cell state *Ct-1* and forget gate *ft*. It then adds the results from the input gate *rt* times *dt*.



- Output gate: Here *ot* is the output gate at time step *t*, and *Wo* and *bo* are the weights and bias for the output gate. The hidden layer *ht* either goes to the next time step or goes up to output as *yt*. In the following code Line 12, yt is obtained by applying another tanh to *ht*. Note that the output gate *ot* is not the output *yt*, it simply is the "gate" to control the output.



LSTM with Regularization

Overfitting is a serious sin in machine learning. When you train a model on your training data and apply it to the test data, the accuracy of the test data usually is less than that of the training data. We know this is because the model has fitted the training data too well, including the noises in the training data. However, if overfitting just makes your prediction for the test data less effective, what's the big deal? Why do academia and practitioners devote decades of work to prevent overfitting?

The real issue is that overfitting not only makes your model inefficient, but it could also make your prediction very wrong. Suppose your final model has ten variables, eight of which capture the real patterns and the other two variables, noises. In other words, the two variables overfit noises and are useless. Suppose you are going to predict new data with your ten-variable model, and suppose the new values for the two variables are large. Guess what will happen? Your prediction will be very wrong due to the two variables and the large values in the new data. So overfitting does not just make your model ineffective, it can make your prediction very wrong.

Deep learning uses the **d**ropout technique to control overfitting. The dropout technique randomly drops or deactivates some neurons for a layer during each iteration. It is like some weights are set to zero. So in each iteration, the model looks at a slightly different structure of itself to optimize the model.

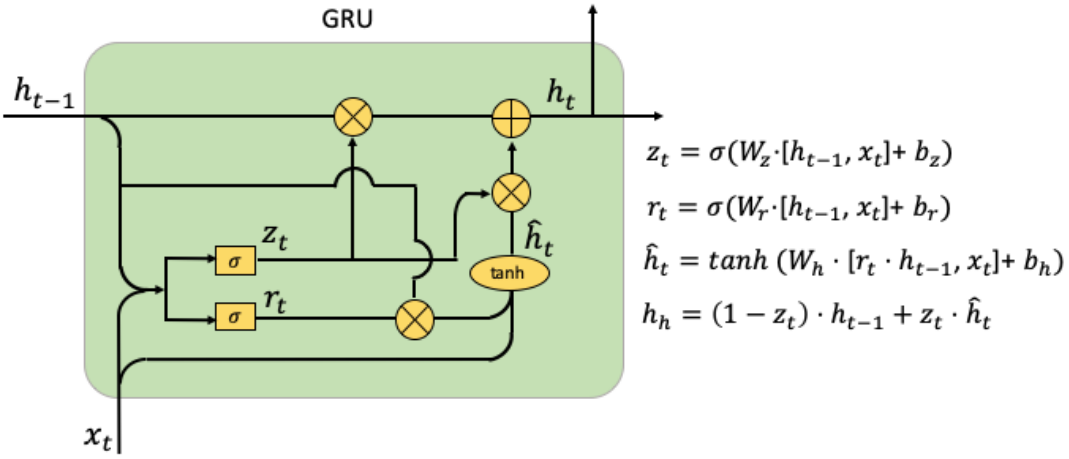
Handling dropout in RNN/LSTM/GRU is also a research topic. In a feedforward neural network, dropping neurons is easier because there is no connectivity between neurons of the same layer. However, in RNN/LSTM/GRU, dropping time steps harms the ability to carry informative signals across time.

D) GRU (Gated Recurrent Units)

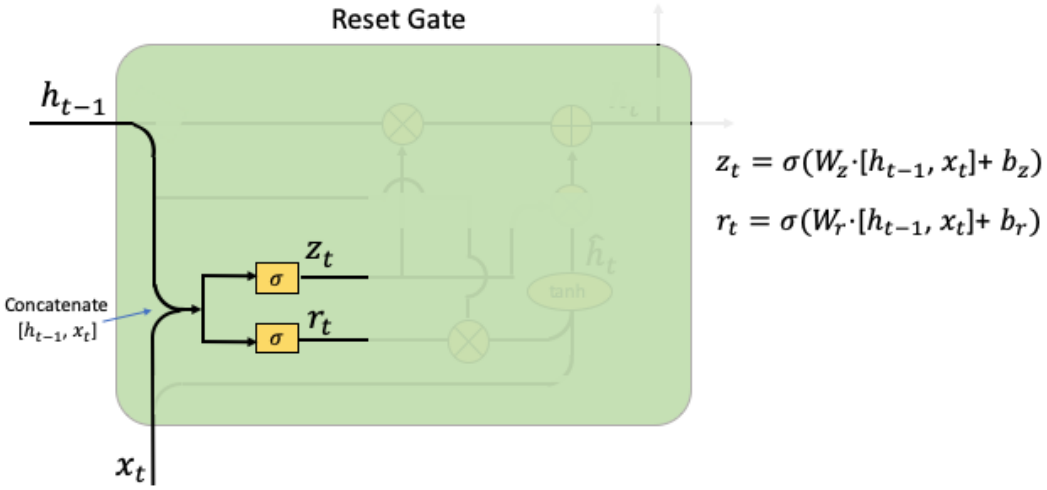
The GRU was invented by Cho et al. (2014) in a company with RNN and LSTM. It is expected more variations of the recursive network will continue to emerge. GRU also aims to solve the vanishing gradient problem. GRU does not have the cell state and the output gate like those in LSTM. It, therefore, has fewer parameters than LSTM. GRU uses the hidden layers to transfer information. GRU calls its two gates the reset gate and the update gate. Let me explain them one by one.

The Structure of GRU:

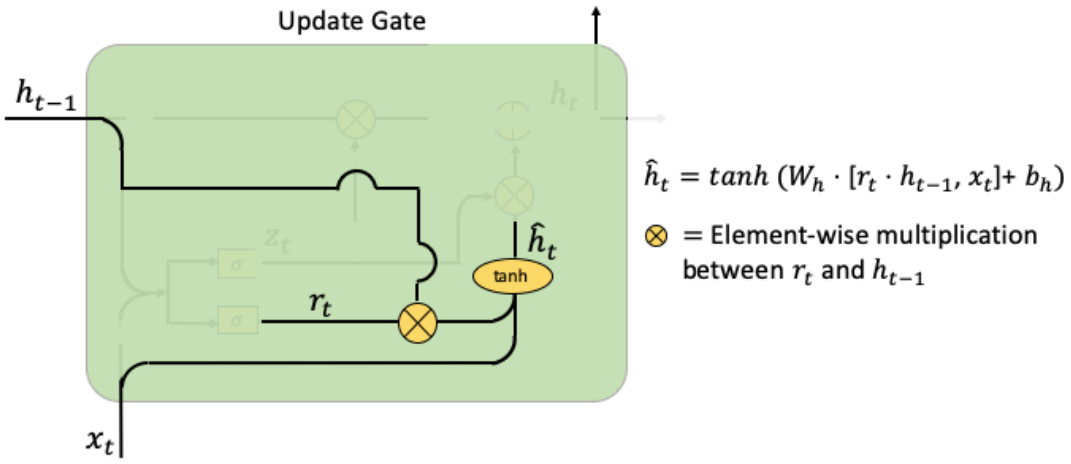
The parameters of GRU include Wr, Wz, and W. The reset signal *rt* determines if the previous hidden state should be ignored while the update signal *zt* determines if the hidden state *ht* should be updated with the new hidden state *hat(ht)*.



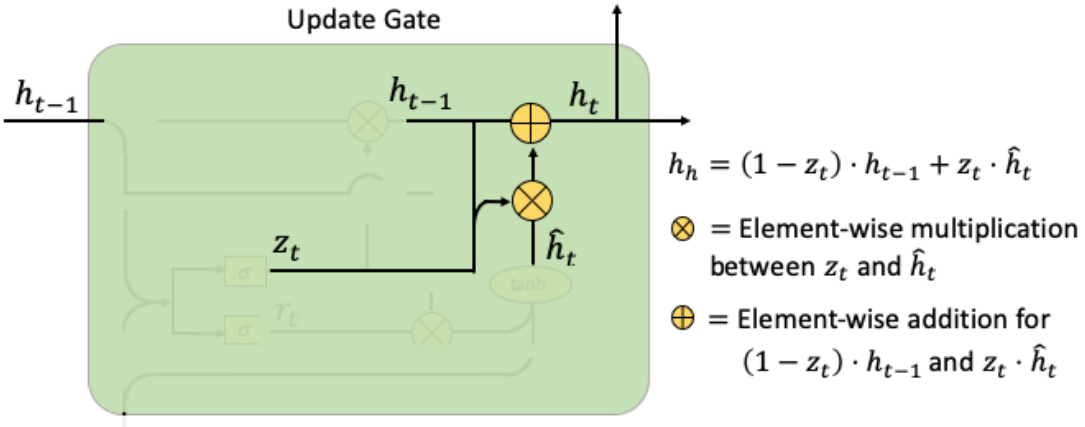
- Reset Gate**:** This achieves what the input gate and forget gate of LSTM achieve. The gate *rt* determines if the previous hidden state should be ignored. The gate *zt* is generated for the update gate with *hat(ht)*. *Wz* and *Wr* are the weight parameters to be trained, *bz* and *br* are the noise vectors.



- Update Gate: (Part 1) This part multiplies *rt* and *ht-1*. The multiplication means how much of ht-1 will be retained or ignored. This creates a temporal *hat(ht)* to be used for the update of *ht*. *Wh* and *bh* are weight parameters and the noise vectors.



- Update Grade: (Part II) This part computes the weighted average between *ht-1* and *hat(ht)*, according to the weight *zt*. If *zt* is close to zero, the past information contributes little and new information contributes more.



5.3 Methodologies

Build a Simple Model

Tensorflow and Keras are the two most popular platforms for modeling neural networks. TensorFlow is an open-source platform for machine learning. It has many libraries and community resources. It enables ML developers to build and deploy ML applications easily. Keras is an open-source deep-learning API written in Python. It supports Tensorflow or Theano. Big companies such as Microsoft, NVIDIA, Google, and Amazon have contributed actively to Keras' growth.
The procedure to build the transfer learning model is the same as building any neural network. It involves three steps in Keras: (1) .sequential() to build the model architecture, (2) .compile () to compile the model with the loss function and optimization method, and (3) .fit() to train the model. The function .sequential() defines the structure of a neural network. It is the pipeline to add any layers to a model. Once a model is defined, the compiling step defines the evaluation metric and the optimizer. The last step executes the actual model training.

(H.1) Declare the Model

This step gives the model specification for a simple RNN model.

(H.2) Compile the Model

The second step of building a neural network is the compiling step. This step defines the evaluation metric and the optimizer. An evaluation metric is the loss function that is used to judge the performance of a model. Keras includes almost all the evaluation metrics. The evaluation metrics belong to three categories: (a) the regression-related metrics, (b) the probabilistic metrics, and © the accuracy metrics. The regression-related metrics include mean squared error (MSE), root mean squared error (RMSE), mean absolute error (MAE), mean absolute percentage error (MAPE), and so on. When your target variable is continuous and you would like to pursue the minimum deviation in terms of percentage errors or absolute errors, you should consider the regression-related metrics.

The **probabilistic metrics** are considered when your prediction is a probability. They include binary cross-entropy and categorical cross-entropy. If your target is binary, you can use binary cross-entropy. If your target is multi-class, categorical cross-entropy.

The **accuracy metrics** calculate how often predictions equal labels. The frequently-used metrics are the accuracy class, binary accuracy class, and categorical accuracy class. As the name suggests, the binary accuracy class calculates how often predictions match binary labels, and the categorical accuracy class calculates how often predictions match multiple labels.

The **optimizer** is a function that optimizes a model. Optimizers use the above loss function to calculate the loss of the model and then try to minimize the loss. Without an optimizer, a machine learning model can’t do anything.

The popular optimizers include the Stochastic Gradient Decent (SGD), RMSprop, Resilient Back Propagation (RProp), the Adaptive Moment (Adam), and the Ada family. The SGD is probably the most widely used optimizer. The RMSprop maintain a moving average of the gradients and uses that average to estimate the variance. The Adam is considered more efficient and requires less memory when working with a large amount of data and parameters. It requires less memory and is efficient. I use rmsprop as the optimizer.

(H.3) Fit the Model

The third step is the training step. This is a time-consuming step. You will see “epoch 1”, and “epoch 2” appearing on your screen.
In this step, we deal with a unique concept in a neural network called “**epoch**”. In an epoch, the model goes through all data exactly once. The model parameters are updated in each epoch until they reach optimal values.
If you specify too many epochs, the model will commit overfitting. It will learn the training data too well but predict poorly for a new dataset. How do we determine the optimal number of epochs? The answers are the loss and accuracy metrics. When a

model is trained with more epochs, the loss will decrease, and the accuracy will increase. After a certain number of epochs, the loss will stop decreasing but increase, and the accuracy decreases. It indicates the model training should end with that epoch.

The model prediction is terrible! Do you know why? This is because the input data were not normalized. In (D.2) we will normalize the input data. **The key takeaway is: that normalization is necessary for RNN/LSTM/GRU**

(1) Normalized Data Are Needed for RNN/LSTM/GRU

When we standardize data to train your model, remember that only the training data are used to fit the scaler transformation, then the scalar is used to transform the test input data. Do not scale x_train and x_test independently. The scalar to convert the scaled predictions to the original scale.

(J) Why Do We Need LSTM/GRU?

The optimizer of RNN gets the first-order derivative of the loss function to search for the optimal values. Because RNN is recursive, the first-order derivation process will make a number smaller and smaller, then eventually vanish. This is called *gradient vanishing*. This certain mathematical process makes RNN not a good choice to retain memories. **We need a recursive structure so that the information does not vanish quickly. This is the motive for LSTM and GRU.** (For readers who may not be familiar with the optimization process: A loss function is a metric that measures the errors between the actual and the predicted values. An optimizer is an algorithm that changes the weights of the neurons to pursue the minimum error. A popular optimizer is the Stochastic Gradient Descent (SGD). The above code specifies RMSprop, Root Mean Square Propagation, as the optimizer.)

(K) Why LSTM (Long Short-Term Memory)?

Hochreiter and Schmidhuber (1997) proposed the LSTM structure to retain memory for RNNs over longer periods. It solves the problem of gradient vanishing (or gradient explosion) by introducing **additional gates, input, and forget gates**. These additional gates can control better over the gradient, enabling what information to preserve and what to forget. These gates are sigmoid functions with output in [0,1] to pass limited information or all information. A value of zero means filtering out the information completely, while a value of one means passing the information completely. The structure is called Long Short-Term Memory because it uses the short-term memory processes to create longer memory. (So do not mislabel "Long **Short-Term** Memory" as "Long-Short Term Memory".)

How does LSTM retain the memory of a long time ago? **LSTM has its layers called the *cell state,* often labeled *Ct,* in addition to the hidden layers** to prevent the old information from vanishing too soon. In our stock price example, the price of this Friday may be influenced by the prices of previous Fridays, or even the price of the same day last year. RNNs may not be able to retain the price information of the same day last year, while LSTM in theory is designed to retain it.

(K.1) Math Explanation for the Issue of Vanishing Gradients

Assume we have a hidden state ht at time step t to make the math simple, we remove the bias *bt* and the input *xt* terms:

$$h_t = \sigma(wh_{t-1})$$

Taking the first-order derivatives we get the following equations. The circled coefficient vector is the key. It will vanish exponentially to zero, or explode exponentially to infinity.

$$\frac{\partial h_{t2}}{\partial h_t} = \prod_{k=1}^{t2-t} w\sigma'(wh_{t2-k})$$

$$\frac{\partial h_{t2}}{\partial h_t} = w^{t2-t} \prod_{k=1}^{t2-t} \sigma'(wh_{t2-k}) \quad \text{Eq. (1)}$$

To overcome the problem, LSTM adds the cell state st. The derivative equation is:

$$\frac{\partial s_{t2}}{\partial s_t} = \prod_{k=1}^{t2-t} \sigma(v_{t2-k}) \quad \text{Eq. (2)}$$

Here vt is the input to the forget gate. Because there is no decaying factor *w*, it does not vanish so fast. However, you may ask that Eq. (2) also has the sigmoid function, so the matrix multiplication also will make the values vanish. You are correct. LSTM still will suffer the problem of vanishing gradients, but not as fast as Eq. (1).

5.4 Use Case Diagram

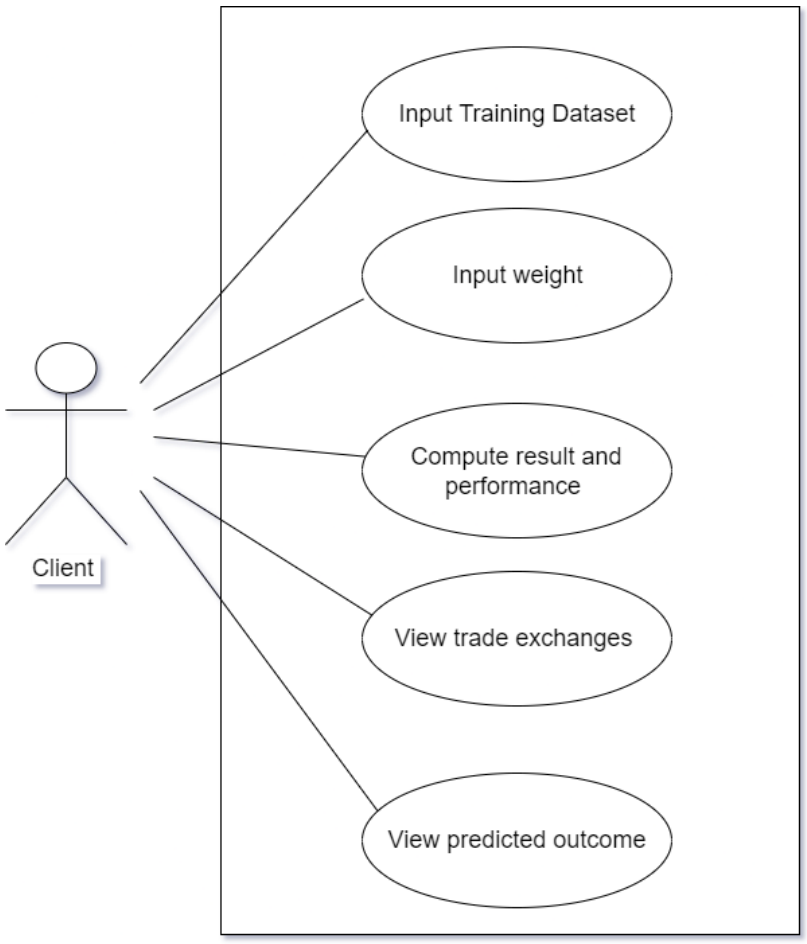


Figure 5.6 : Use Case Diagram

The case diagram of the proposed system is shown in fig. The proposed system allows the user to provide the training data and the input weight to achieve its result and to calculate the performance. Training Neural network requires initialization of input weights and the first input weight will be adjusted until the optimal prediction is achieved.

A use case diagram at its simplest is a representation of a user's interaction with the system that shows the relationship between the user and the different use cases in which the user is involved. A use case diagram can identify the different types of users of a system and the different use cases and will often be accompanied by other types of diagrams as well. The use cases are represented by either circles or ellipses.

5.5 Data Flow Diagram.

A Data Flow Diagram (DFD) is a visual representation of how data flows within a system. For a stock market prediction system, the DFD can illustrate how data is

collected, processed, and utilized to generate predictions. Here's an example of a simple DFD for a stock market prediction system:

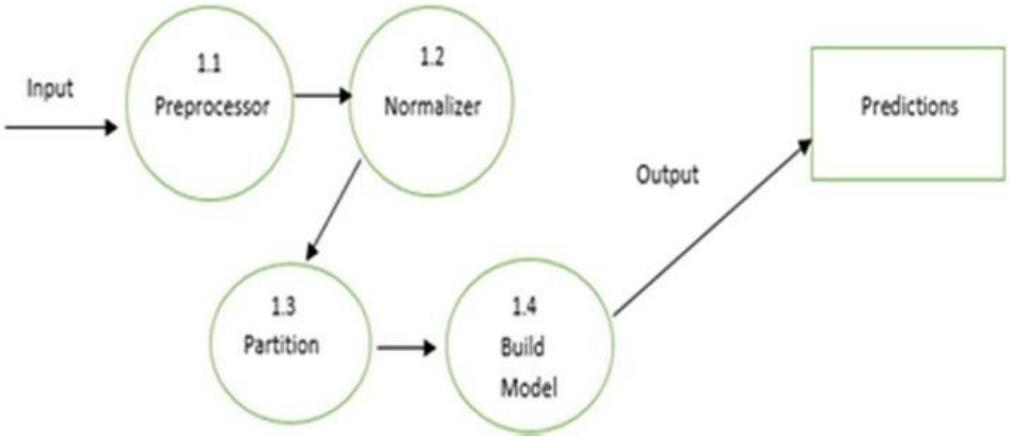


Figure 5.7 : Data Flow Diagram

Description of processes:

Data Input: Collects raw data related to stock markets, such as historical prices, news articles, financial reports, and social media sentiment.

Data Preprocessing: Cleans and transforms the raw data into a suitable format for analysis. This process may involve tasks like data cleansing, normalization, feature selection, and data integration.

Model Training: Uses the preprocessed data to train a predictive model. This process typically involves applying machine learning or statistical techniques to build a model that can learn patterns from historical data.

Data Prediction: Applies the trained model to new or unseen data to generate stock market predictions. This process utilizes the learned patterns and makes predictions based on the input data.

Data Output: Presents the predicted results, such as stock price forecasts, market trends, or investment recommendations, in a user-friendly format. The output can be displayed through a web interface, mobile app, or other means.

5.6 Entity Relationship Diagram

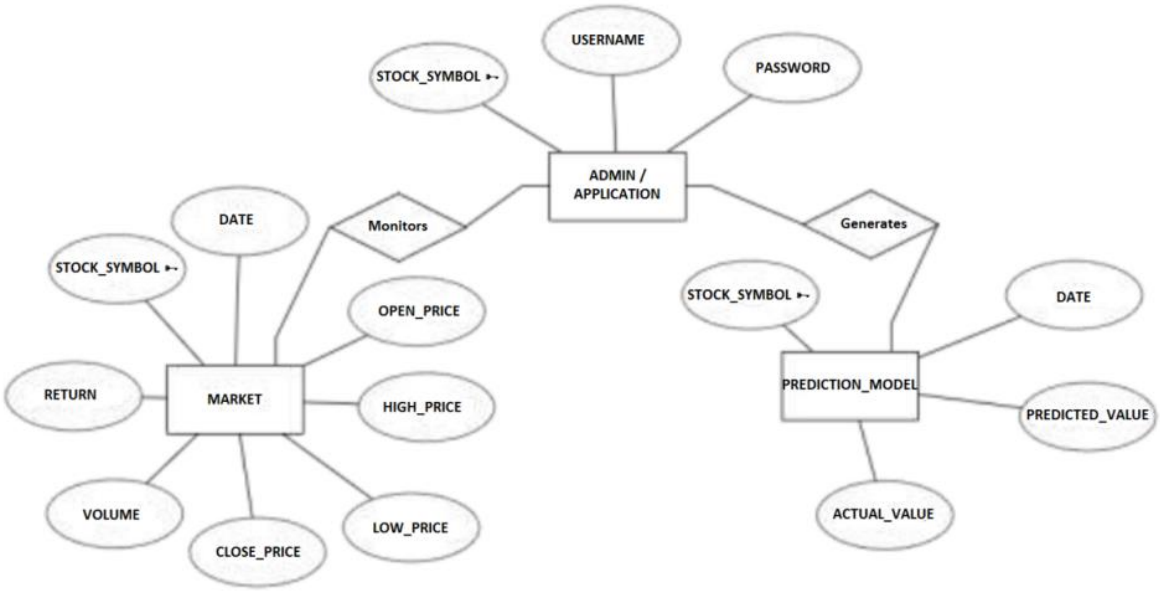


Figure 5.8 Entity Relationship Diagram

The Entity-Relationship (ER) model is a conceptual modeling technique used to represent the entities, relationships, and attributes within a system. It helps to visualize and understand the structure and dynamics of a system.

In the context of a stock price prediction system, an ER diagram can be used to represent the entities involved and their relationships. Here is a general overview of the key components of an ER diagram:

1. Entities: Entities represent the real-world objects or concepts that are relevant to the system. In a stock price prediction system, entities could include "Stock" and "Prediction." Entities are typically represented as rectangles in an ER diagram.

2. Attributes: Attributes describe the properties or characteristics of an entity. They provide additional information about the entities. For example, attributes of the "Stock" entity could include "stock_id," "name," and "symbol." Attributes are typically depicted within the entity rectangle.

3. Relationships: Relationships illustrate the associations between entities. They define how entities are related or connected to each other. In a stock price prediction system, there is likely a relationship between the "Stock" and "Prediction" entities, indicating that each prediction is associated with a specific stock. Relationships are typically represented as lines connecting the related entities.

4. Cardinality and Multiplicity: Cardinality describes the number of occurrences of an entity that can be associated with another entity through a relationship. It helps define the nature of the relationship, such as one-to-one, one-to-many, or many-to-many. Multiplicity defines the specific numbers or ranges of occurrences allowed in the relationship.

By combining these components, an ER diagram provides a visual representation of the stock price prediction system, showcasing the entities, their attributes, and the

relationships between them. The diagram helps in understanding the structure, dependencies, and constraints of the system.

It's important to note that the specific design of an ER diagram for a stock price prediction system may vary depending on the requirements and scope of the project. The diagram can be expanded and refined to include additional entities, attributes, and relationships as needed to accurately represent the system.

5.7 Activity Diagram of the System.

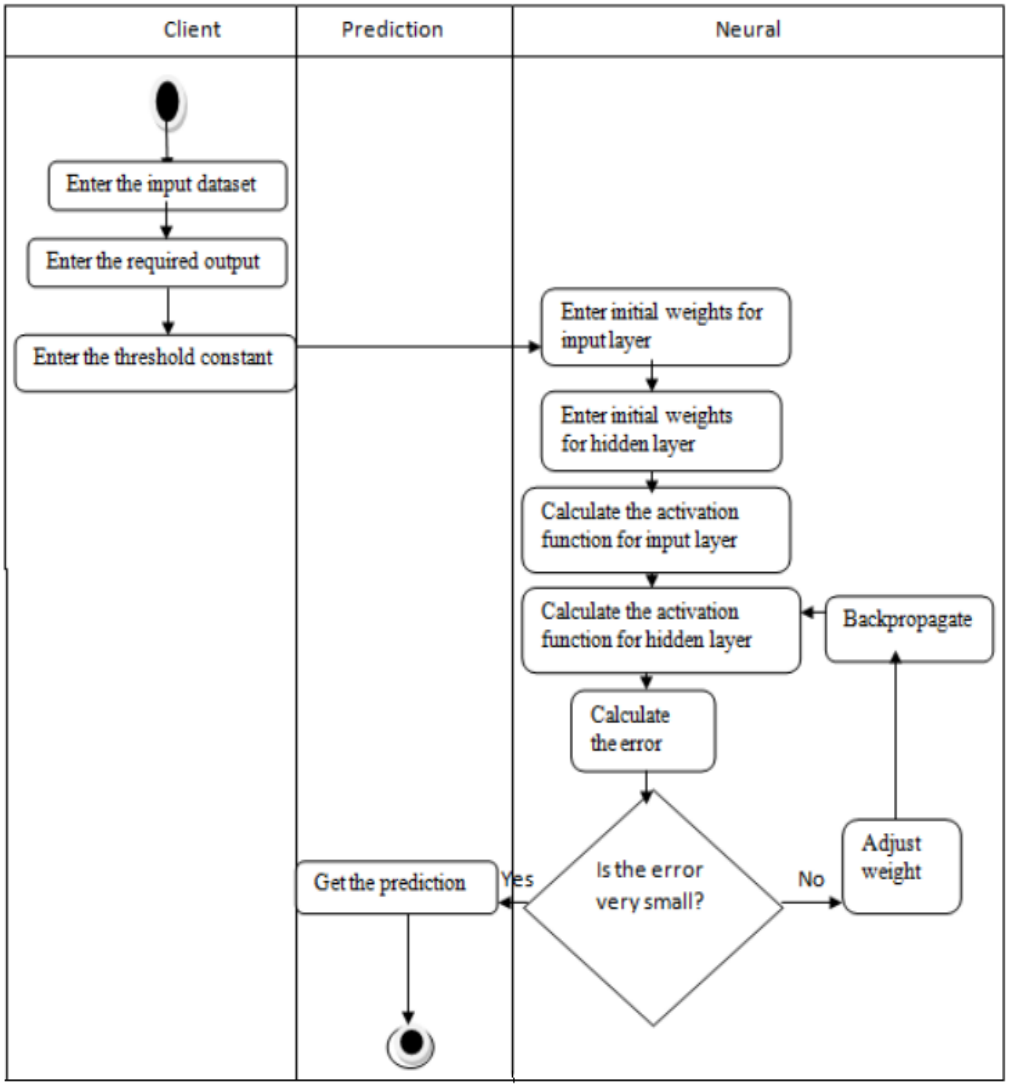


Figure 5.9 Activity Diagram

An activity diagram is essentially flowchart, showing flow of controls from a one activity to another. Unlike a traditional flowchart, it can model the dynamic functional view of a system. An activity diagram represents an operation on some classes in the system that results to changes in the state of the system. From the diagram shown in fig, the client is expected to provide the input dataset and the required output. The required output is used in back propagation, for the system uses it to compare its predicted value from time to time in other to get the optimal prediction. The client selects a threshold constant and looking at the input value the neural network initializes the weight for the input layer and the hidden layer. Then the calculation for the activation function of both the input and the output layer is done and the system calculates the error. If the is large then the system adjust the weight and back propagate it to the input layer for further calculation to be done. The system outputs its prediction whenever the error is small.

Activity diagrams are graphical representations of workflows of step wise activities and actions with support for choice, iteration, and concurrency. In the Unified Modelling Language, activity diagrams are intended to model both computational and organizational processes (i.e., workflows), as well as the data flows intersecting with the related activities.

5.8 Sequence Diagram of the System

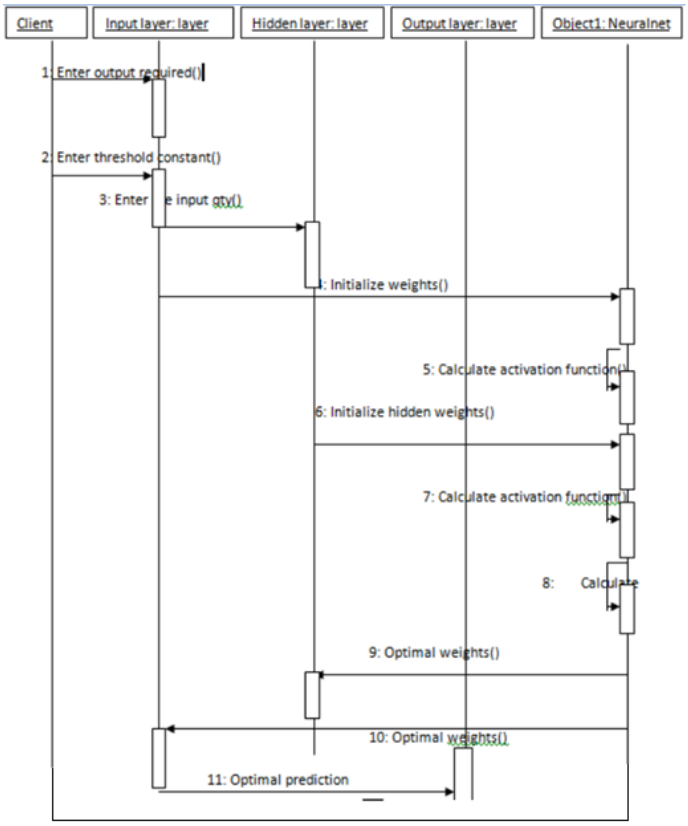


Figure 5.10 Sequence Diagram

Fig, shows the sequence diagram of the proposed system. The sequence diagram number the actions starting from the data input to the optimal prediction. There is an arrow direction to show the sequence of flow for the action taking to arrive at the optimal prediction.

A sequence diagram shows object interactions arranged in time sequence. It depicts the objects involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario. Sequence diagrams are typically associated with use case realizations in the Logical View of the system under development. Sequence diagrams are sometimes called event diagrams or event scenarios.

CHAPTER 6

SOFTWARE INFORMATION

Python

Python is an interpreted, high-level and general-purpose programming language. Created by Guido van Rossum and first released in 1991, Python's design philosophy emphasizes code readability with its notable use of significant whitespace. Its language constructs and object-oriented approach aim to help programmers write clear, logical code for small and large-scale projects.

Python is dynamically typed and garbage-collected. It supports multiple programming paradigms, including structured (particularly, procedural), object-oriented, and functional programming. Python is often described as a "batteries included" language due to its comprehensive standard library

Python was created in the late 1980s as a successor to the ABC language. Python 2.0, released in 2000, introduced features like list comprehensions and a garbage collection system with reference counting. Python 3.0, released in 2008, was a major revision of the language that is not completely backward-compatible, and much Python 2 code does not run unmodified on Python 3. The Python 2 language was officially discontinued in 2020 (first planned for 2015), and "Python 2.7.18 is the last Python 2.7 release and therefore the last Python 2 release."[30] No more security patches or other improvements will be released for it. With Python 2's end-of-life, only Python 3.6.x and later are supported.

Python interpreters are available for many operating systems. A global community of programmers develops and maintains CPython, a free and open-source reference implementation. A non-profit organization, the Python Software Foundation, manages and directs resources for Python and CPython development.

Python was conceived in the late 1980s by Guido van Rossum at Centrum Wiskunde Informatica (CWI) in the Netherlands as a successor to the ABC language (itself inspired by SETL), capable of exception handling and interfacing with the Amoeba operating system. Its implementation began in December 1989. Van Rossum shouldered sole responsibility for the project, as the lead developer, until 12 July 2018, when he announced his "permanent vacation" from his responsibilities as Python's Benevolent Dictator For Life, a title the Python community bestowed upon him to reflect his long-term commitment as the project's chief decision-maker. He now shares his leadership as a member of a five-person steering council. In January 2019, active Python core developers elected Brett Cannon, Nick Coghlan, Barry Warsaw, CarolWilling and Van Rossum to a five-member "Steering Council" to lead the project.

Anaconda

Anaconda is a free and open-source distribution of the Python and R programming languages for scientific computing (data science, machine learning applications, large-scale data processing, predictive analytics, etc.), that aims to simplify package management and deployment. The distribution includes data-science packages suitable for Windows, Linux, and macOS. It is developed and maintained by Anaconda, Inc., which was founded by Peter Wang and Travis Oliphant in 2012. As an Anaconda, Inc. product, it is also known as Anaconda Distribution or Anaconda Individual Edition, while other products from the company are Anaconda Team Edition and Anaconda Enterprise Edition, both of which are not free.

Package versions in Anaconda are managed by the package management system conda. This package manager was spun out as a separate open-source package as it ended up being useful on its own and for other things than Python. There is also a small, bootstrap version of Anaconda called Miniconda, which includes only conda, Python, the packages they depend on, and a small number of other packages. Anaconda distribution comes with over 250 packages automatically installed, and over 7,500 additional open-source packages can be installed from PyPI as well as the conda package and virtual environment manager. It also includes a GUI, Anaconda Navigator, as a graphical alternative to the command line interface (CLI).

The big difference between conda and the pip package manager is in how package dependencies are managed, which is a significant challenge for Python data science and the reason conda exists. When pip installs a package, it automatically installs any dependent Python packages without checking if these conflict with previously installed packages[citation needed]. It will install a package and any of its dependencies regardless of the state of the existing installation. Because of this, a user with a working installation of, for example, Google Tensorflow, can find that it stops working having used pip to install a different package that requires a different version of the dependent numpy library than the one used by Tensorflow. In some cases, the package may appear to work but produce different results in detail. In contrast, conda analyses the current environment including everything currently

installed, and, together with any version limitations specified (e.g. the user may wish to have Tensorflow version 2,0 or higher), works out how to install a compatible set of dependencies, and shows a warning if this cannot be done.

Open source packages can be individually installed from the Anaconda repository, Anaconda Cloud (anaconda.org), or the user's own private repository or mirror, using the conda install command. Anaconda, Inc. compiles and builds the packages available in the Anaconda repository itself, and provides binaries forWindows 32/64 bit, Linux 64 bit and MacOS 64-bit. Anything available on PyPI may be installed into a conda environment using pip, and conda will keep track of what it has installed itself and what pip has installed. Custom packages can be made using the conda build command, and can be shared with others by uploading them to Anaconda Cloud, PyPI or other repositories. The default installation of Anaconda2 includes Python 2.7 and Anaconda3 includes Python 3.7. However, it is possible to create new environments that include any version of Python packaged with conda.

CHAPTER 7

PROJECT PLAN

In the project plan section we are showing the overall things in our project we are performing untill now. How much time it took to complete processes like Introduction and problem statement , Study of literature survey, Project statement, Software requirement and specification, Data gathering and preprocessing task such as data cleaning, feature extraction, normalization of data, visualize data to studie more in inside view and trande of stock market, Train and testing models to fing accuracy and loss in each model, consistency in model. Comparing the models to get the better one.

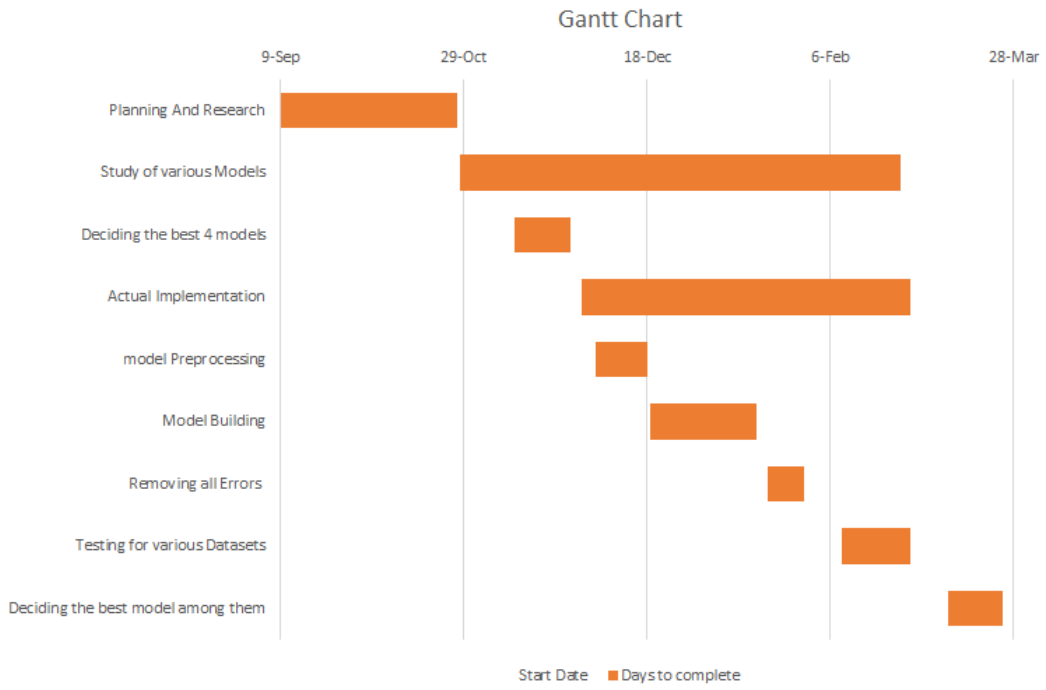


Figure : 7.1 Gantt chart

CHAPTER 8

TEST CASES

Sr. No	Description	Test Case I/P	Actual Result	Expected	Test Criteria (P/F)
1	Install Python	Python Exe	Should get install properly	Proper intalled	Pass
2	Install jupyter	Anaconda navigator exe	Install jupyter notebook	Run jupyter notebook	Pass
3	Installing Libraries	Pip install	Libraries install	Library installed successfully	Pass
4	Import data	Use pandas lib.	Data imported	Fetch data for model	Pass
4	Normalize Data	Dataset scaling	Data in range	Normalize values	Pass
5	Visualize Data	Data to be study	Graphs and plots of data	Colorful graph to examine	Pass
6	Model Building	Various mdoels to be build	Algorithms applied	It predicts the future data	Pass
7	Result Compare	Study of models	Calculated the Best one	Finding Best Model	Pass

Table 8.1 Test Cases

CHAPTER 9

ANALYSIS CONCLUSION

Analysis

There are many research paper that deals with the forecasting of stock market. But we observed that many of them perform the study on machine learning algorithms like regression algorithm, support vector machine algorithm, random forest regression algorithm etc. So we decide to perform the operation on deep learning algorithm and do the study on the result by comparing the result. Here we use the two machine learning and two deep learning algorithms. Combine these four algorithms namely linear regression algorithm, random forest regressor, long short term model (LSTM) and Gated recurrent unit (GRU).

Let's discuss the result we obtain

1. Linear Regression:

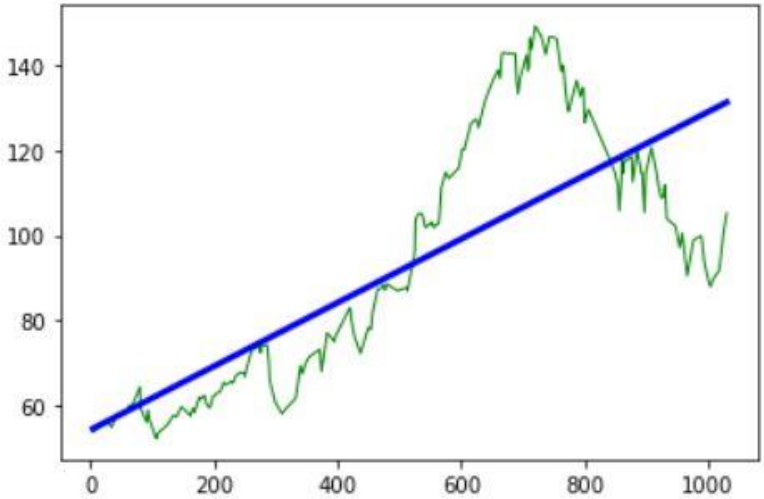


Fig: 9.1 Linear Regression

Linear Regression is a basic machine learning algorithm used for regression tasks. In this algorithm, a linear relationship is assumed between the independent variable (X) and the dependent variable (Y). The algorithm fits a straight line through the data points to predict the value of Y for a given X. The graph for Linear Regression shows a straight line representing the best fit line through the data points. The accuracy of the model depends on how well the data points are distributed around the line.

2. Random Forest Regressor:

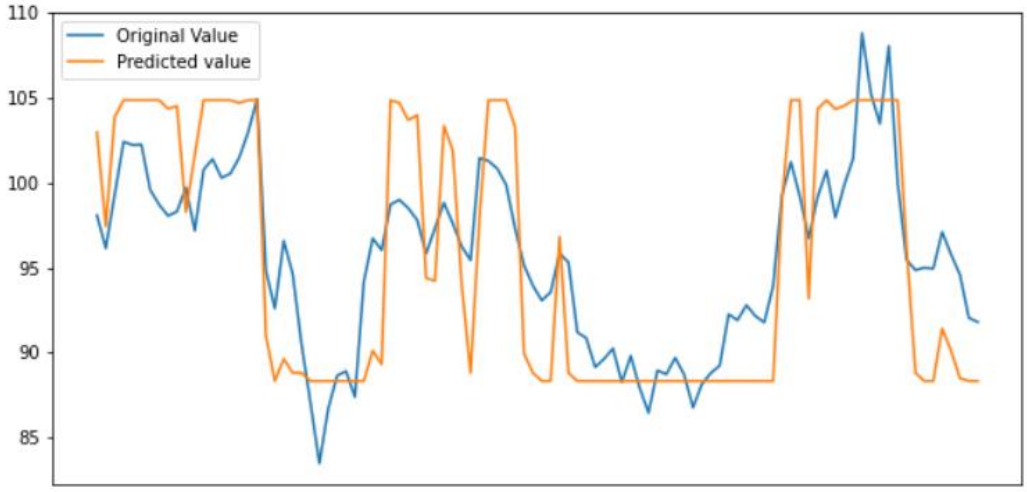


Fig: 9.2 Random Forest

Random Forest is an ensemble learning algorithm used for regression and classification tasks. It creates multiple decision trees and combines them to improve the accuracy and reduce overfitting. The algorithm randomly selects a subset of features and data samples to create each tree. The graph for Random Forest shows a curve that captures the non-linear relationship between the independent and dependent variables. The accuracy of the model depends on the number of decision trees in the forest and the hyperparameters of the algorithm.

3. LSTM:

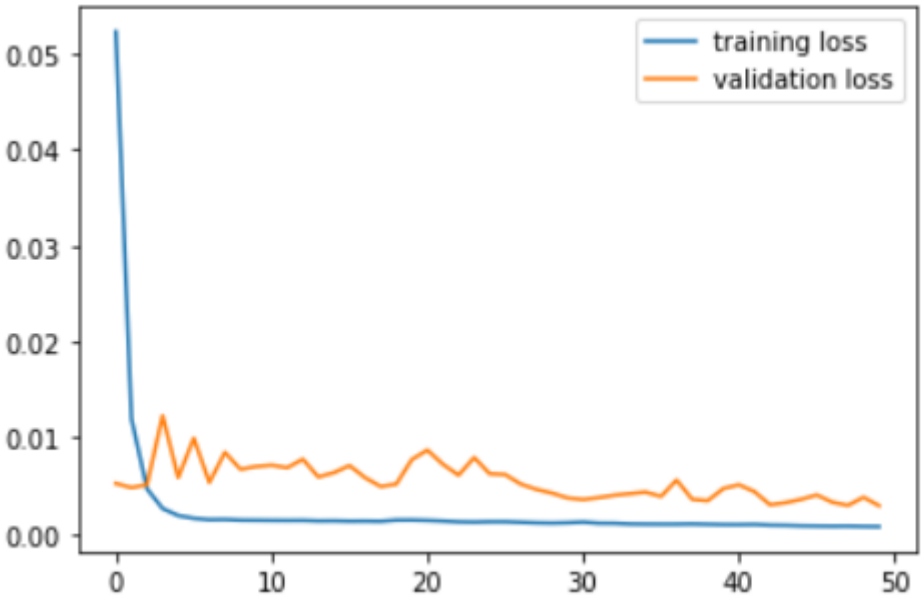


Fig. 9.3 Loss Iteration by LSTM

Long Short-Term Memory (LSTM) is a deep learning algorithm used for sequence prediction tasks. It is a type of Recurrent Neural Network (RNN) that can remember past information and use it to make future predictions. The algorithm is suitable for time series data like stock market prediction. The graph for LSTM shows a curve that

captures the trends and patterns in the data. The accuracy of the model depends on the architecture of the LSTM network, the hyperparameters, and the amount of training data.

4. GRU:

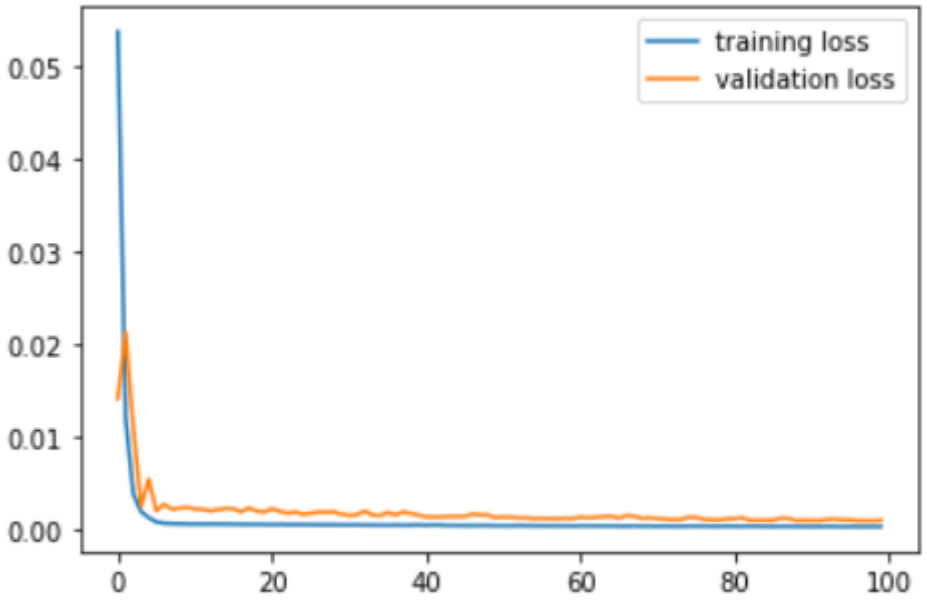


Fig 9.4 Loss iteration by GRU

Gated Recurrent Unit (GRU) is another deep learning algorithm used for sequence prediction tasks. It is similar to LSTM but has fewer parameters, which makes it faster to train and less prone to overfitting. The algorithm uses gates to control the flow of information and decide what to forget and what to remember. The graph for GRU shows a curve that captures the trends and patterns in the data. The accuracy of the model depends on the architecture of the GRU network, the hyperparameters, and the amount of training data.

In conclusion, the choice of the best model for stock market prediction depends on various factors, including the type and amount of data, the complexity of the problem, and the performance metrics used for evaluation. Therefore, it is essential to experiment with different models and analyze their results to choose the most suitable one for the specific task at hand. From the study we observed that the our two deep learning algorithm perform better than the machine learning algorithms and give us the best accuracy.

Conclusion

The project aims to forecasting the trends in stock market using deep learning and machine learning technologies for retail investors, connecting predictions made by machine learning models to retail investor. It helps investors navigate through the stock market with additional analysis and help them make more informed decision. The findings demonstrated that the application provides significance in trend prediction. When compared to the baseline, the prediction shows useful trend tendency with the real stock trend. The models used in the application will continue to improve itself by searching for a better model topology, structure and hyper parameters through evolution algorithm. The findings concluded the usefulness of evolution algorithm in lowering the mean squared error when predicting stock prices, which is helpful for improving the trend prediction for retail investors.

The aim of creating an user-friendly system for retail investors whom does not have previous technical knowledge to navigate the machine model predictions result with useful benchmarks. With better presentation of stock price predictions could be developed to help investors understand the implications of the stock price predictions, e.g. when to buy or sell. This would allow investors to make more informed decisions based on the Deep learning models and truly democratize Deep learning and machine learning technologies, which were believed to be only in the hands of very few people.

CHAPTER 10

REFERENCES

[1] Shrinath ravikumar, Prasad saraf "Prediction of Stock Prices using Machine Learning (Regression,Classification) Algorithms", in 2020 International Conference for Emerging Technology (INCET) Belgaum, India. Jun 5-7, 2020

[2] D. M. Q. Nelson, A. C. M. Pereira and R. A. de Oliveira, "Stock market's price movement prediction with LSTM neural networks," 2017 International Joint Conference on Neural Networks (IJCNN), Anchorage, AK, USA, 2017, pp. 1419-1426, doi: 10.1109/IJCNN.2017.7966019.

[3] M. Nabipour, P. Nayyeri, H. Jabani, S. S. and A. Mosavi, "Predicting Stock Market Trends Using Machine Learning and Deep Learning Algorithms Via Continuous and Binary Data; a Comparative Analysis," in IEEE Access, vol. 8, pp. 150199-150212, 2020, doi: 10.1109/ACCESS.2020.3015966.

[4] Mehta, Yash & Malhar, Atharva & Shankarmani, Radha. (2021). Stock Price Prediction using Machine Learning and Sentiment Analysis. 1-4. 10.1109/INCET51464.2021.9456376.

[5] Ishita Parmar, Navanshu Agarwal, Sheirsh Saxena, Ridam Arora, Shikhin Gupta, Himanshu Dhiman, Lokesh Chouhan "Market Prediction Using Machine Learning" , "Stock market prediction using machine learning techniques," 2016 3rd International Conference on Computer and Information Sciences (ICCOINS), Kuala Lumpur, 2016, pp. 322-327.

[6] Pahwa, Kunal & Agarwal, Neha. (2019). Stock Market Analysis using Supervised Machine Learning. 197-200. 10.1109/COMITCon.2019.8862225.

[7] M. Usmani, S. H. Adil, K. Raza and S. S. A. Ali, "Stock market prediction using machine learning techniques," 2016 3rd International Conference on Computer and Information Sciences (ICCOINS), Kuala Lumpur, Malaysia, 2016, pp. 322-327, doi: 10.1109/ICCOINS.2016.7783235.

[8] Lin, Yaohu & Liu, Shancun & Yang, Haijun & Wu, Harris. (2021). Stock Trend Prediction Using Candlestick Charting and Ensemble Machine Learning Techniques With a Novelty Feature Engineering Scheme. IEEE Access. 9. 101433-101446. 10.1109/ACCESS.2021.3096825.

[9] L. Chen, Z. Qiao, M. Wang, C. Wang, R. Du and H. E. Stanley, "Which Artificial Intelligence Algorithm Better Predicts the Chinese Stock Market?," in IEEE Access, vol. 6, pp. 48625-48633, 2018, doi: 10.1109/ACCESS.2018.2859809.

[10] Yahoo finance data - https://in.finance.yahoo.com/

[11] Paper publication site - https://ijarsct.co.in/

[12] https://www.kaggle.com/

[13]https://medium.com/swlh/a-technical-guide-on-rnn-lstm-gru-for-stock-price-prediction-bce2f7f30346

APPENDDICES

APPENDIX 1

PLAGIARISM REPORT

PLAGARISM REPORT

Identical Words	50
Words with Minor Changes	118
Paraphrased Words	197
Omitted Words	5842

Figure: 11.1 Plagarism Report

APPENDIX 2

BASE PAPER