وزارة التعليم العالي والبحث العلمي

MINISTRY OF HIGHER EDUCATION
AND SCIENTIFIC RESEARCH

Smart Trading Platform and Stock Price Prediction Using Deep Learning

A Project report submitted in partial fulfillment of the requirements for the award of the degree of

BACHELOR OF TECHNOLOGY
IN
ELECTRONICS AND COMMUNICATIONS ENGINEERING

By:

Ahmed Elsaid Slama - Ahmed Mohamed Elsherbeny - Ahmed Abdelkreem Mohamed - Salaheldin Atef Dahroug - Mohamed Tharwat Sheta - Hossameldin Mohamed Elkallaf - Mohamed Khater Ibrahim - Rashad Elsayed Elmahdy - Amr Labeb Shokry - Mohamed Aboelyazid Aboaish - Mahmoud Elqazaz.

Supervised By:

Assist. Prof. Dr. Warda Mohammed

Assist. Lect. Ghada Abdellatif

DEPARTMENT OF ELECTRONICS AND COMMUNICATIONS ENGINEERING.

Nile Higher Institute For Engineering And Technology.

2023

Smart Trading Platform and Stock Price Prediction Using Deep Learning

A Project report submitted in partial fulfillment of the requirements for the award of the degree of

BACHELOR OF TECHNOLOGY
IN
ELECTRONICS AND COMMUNICATIONS ENGINEERING

Submitted by: **Ai Candle Team**.

Under supervision of:

Assist. Prof. Dr. Warda Mohammed

Assist. Lect. Ghada Abdellatif

DEPARTMENT OF ELECTRONICS AND COMMUNICATIONS ENGINEERING.

Nile Higher Institute For Engineering And Technology.

2019-2023

Contents

Team Structure

01

Team specialized in Deep learning and Unsupervised ML.

- Ahmed Elsaid Slama.
- Ahmed Abdelkreem Mohamed.
- Ahmed Mohamed Elsherbeny.

02

Team specialized in Web Scraping (Real time Data).

- Salaheldin Atef Dahroug.
- Mohamed Tharwat Sheta.

03

Team specialized in Mobile application Software UI/UX.

- Mohamed Aboelyazid Aboaish.
- Mahmoud Elqazaz.
- Amr Labeb Shokry.
- Rashad Elsayed Elmahdy.

04

Team specialized in gathering information's about stock market.

- Mohamed Khater Ibrahim.
- Hossameldin Mohamed Elkallaf.

Acknowledgements

First of all, Praise be to Allah, the Cherisher and Sustainer of the worlds for his constant love and success for us.

It is a great honor to express my deep gratitude to Assist. Prof. Dr. Warda Mohammed for her remarkable supervision and continuous encouragement. With her advice and care during this work.

Also, we are very grateful and truly indebted to Assist. Lect. Ghada Abdellatif who's not only served as my supervisor but also for her encouragement, valuable guidance, and indispensable help. Her words of advice, his trust, and his patience and understanding helped us to fulfill this work.

AI Candle Team

2023

List of Publications

- **W. Shaban, A. Elsaid, and AI candle team,"*SMP-DL: A Novel Stock Market Prediction Approach based on Deep Learning for Effective Trend Forecasting,”* Neural Computing and Application, Springer, Under Review (IF 5.5).**

List of Figures

List of Tables

List of Symbols

LIST OF SYMBOLS

Symbol	Indication
X_t	Input at current state
X_{t-1}	Input at Previous state
C_t	Current Cell State
C_{t-1}	Previous Cell State
h_t	Current hidden/output State
h_{t-1}	Previous hidden/output State
σ	Sigmoid Function
$tanh$	Hyperbolic tangent function

List of Abbreviations

Abbrev.	Full Caption
NYSE	The New York Stock Exchange.
U.S.	United States.
IBM	International Business Machines Corporation.
AI	Artificial Intelligence.
ML	Machine Learning.
DL	Deep Learning.
RNN	Recurrent Neural Network.
LSTM	Long Short Term Memory.
GRU	Gate Recurrent Unit.
BI-GRU	Bi-Directional Gate Recurrent Unit.
ANN	Artificial Neural Network.
DNN	Deep Neural Network.

Abstract

As the economy has grown rapidly in recent years, more and more people have begun putting their money into the stock market. Thus, predicting trends in the stock market is regarded as a crucial endeavor, and one that has proven to be more fruitful than others. Profitable investments will result in rising stock prices. Investors face significant difficulties making stock market-related predictions due to the lack of movement and noise in the data. In this Project a new system for predicting stock market price is introduced namely, Stock Market Prediction based on Deep Leaning (SMP-DL/ (**IEEE 2941-2021**)). Also, in this Project, a mobile application based on SMP system was designed for anyone to have it on their mobile. The design application called Smart Trading Platform (STP). The main aim of this application is to enable anyone to speculate in the stock exchange in a simple and secure way. Firstly, SMP-DL/ (**IEEE 2941-2021**) splits into two stages which are; (I) Data Preprocessing (DP), and (ii) Stock Price's Prediction (SP^2). In the first stage, data is preprocessed to obtain cleaned ones through several stages which are; detect and reject missing value, reject outlier items, feature selection, and data normalization. Then in the second stage (e.g., SP^2), the cleaned data will pass through the used predicted model. In SP^2, Long Short-Term Memory (LSTM) combined with Bidirectional Gated Recurrent Unit (BiGRU) to predict the closing price of stock market. The obtained results showed that the proposed system perform well when compared to other existing methods. As RMSE, MSE, MAE, and R2 values are 0.2883, 0.0831, 0.2099, and 0.9948. Moreover, the proposed method was applied using different datasets and it performs well.

Keywords: *financial, Stock Price, Stock market, forecasting, Deep Learning, ML, DL, RNN, LSTM, GRU, BI-GRU.*

Problem Statement

- **Market Risk.**

- **Heavy loss:**
  - **One of the biggest causes of loss is the pressure on the Internet at the time of market activity.**

  - **The use of low-efficiency platform, and poor market analysis.**

- **Complexity and dynamic environment of stock market.**

Solutions:

- **We built a Deep Learning model using RNN to predict the movement of stocks in the future and achieve the highest profit return, in addition to we provided a safe mobile application platform for trading.**

Chapter 1

Introduction

About Stockmarket.

Introduction

It's no secret that the financial market is fascinating to students and scholars of all stripes because it's a dynamic and ever-changing environment wherein to practice and hone their skills as traders, analysts, and other professionals. While different investors may approach the market from different angles for example, by studying market behavior, identifying influential factors, trading stocks, forecasting the direction of the market, making asset recommendations for portfolio management, etc. a lack of financial literacy and understanding of basic economic principles can have a significant impact on investment returns [1]. In the world, there are numerous stock exchanges that make up the stock market or equity market. Due to the law of supply and demand, investors and the general public buy and sell shares whose prices are constantly fluctuating [2,3]. Owning stock or shares in a company gives you ownership in it. A share is sought after at the lowest cost by buyers, while a share is sought after at the highest cost by sellers.

The term stock market as shown in Fig.1 refers to several exchanges in which shares of publicly held companies are bought and sold. Such financial activities are conducted through formal exchanges and via Over-The-Counter (OTC) marketplaces that operate under a defined set of regulations. Both "stock market" and "stock exchange" are often used interchangeably. Traders in the stock market buy or sell shares on one or more of the stock exchanges that are part of the overall stock market. Fig.2 shows stock exchange operators worldwide as of March 2023, by market capitalization of listed companies.

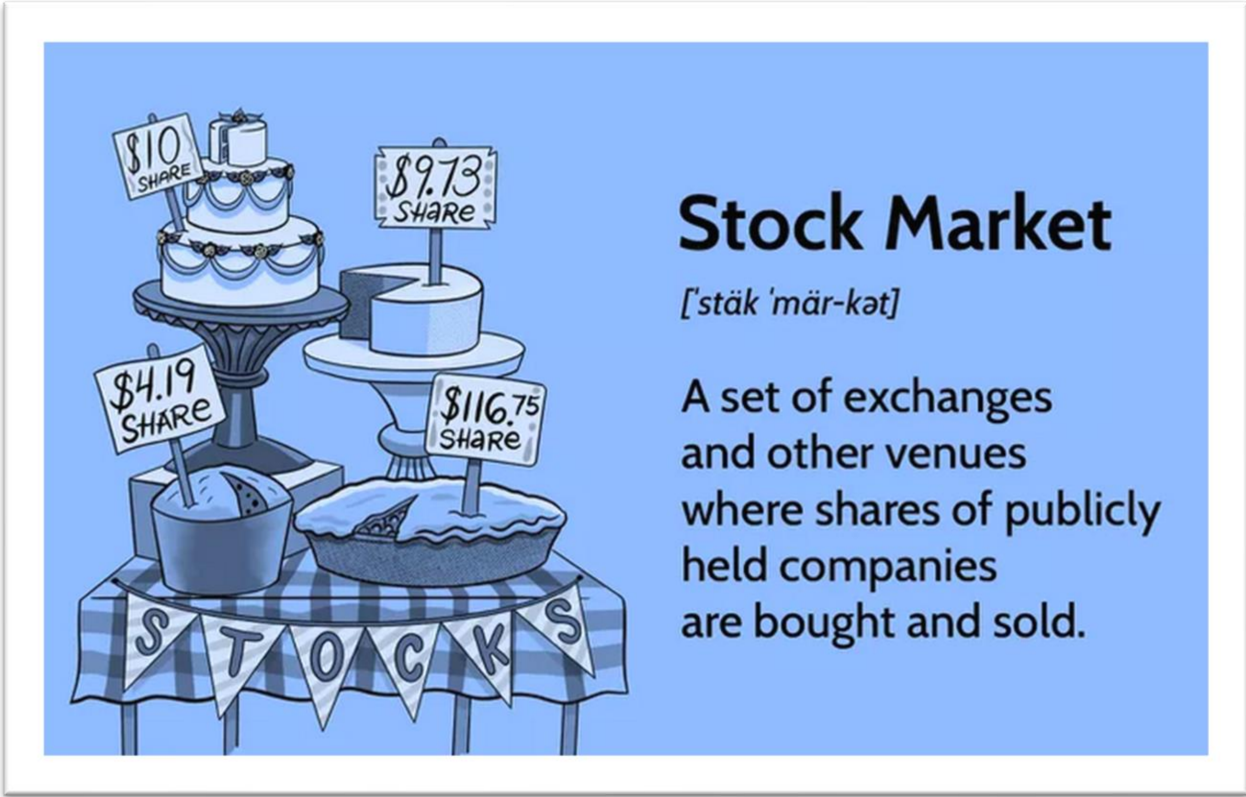

Fig.1. stock market explanation

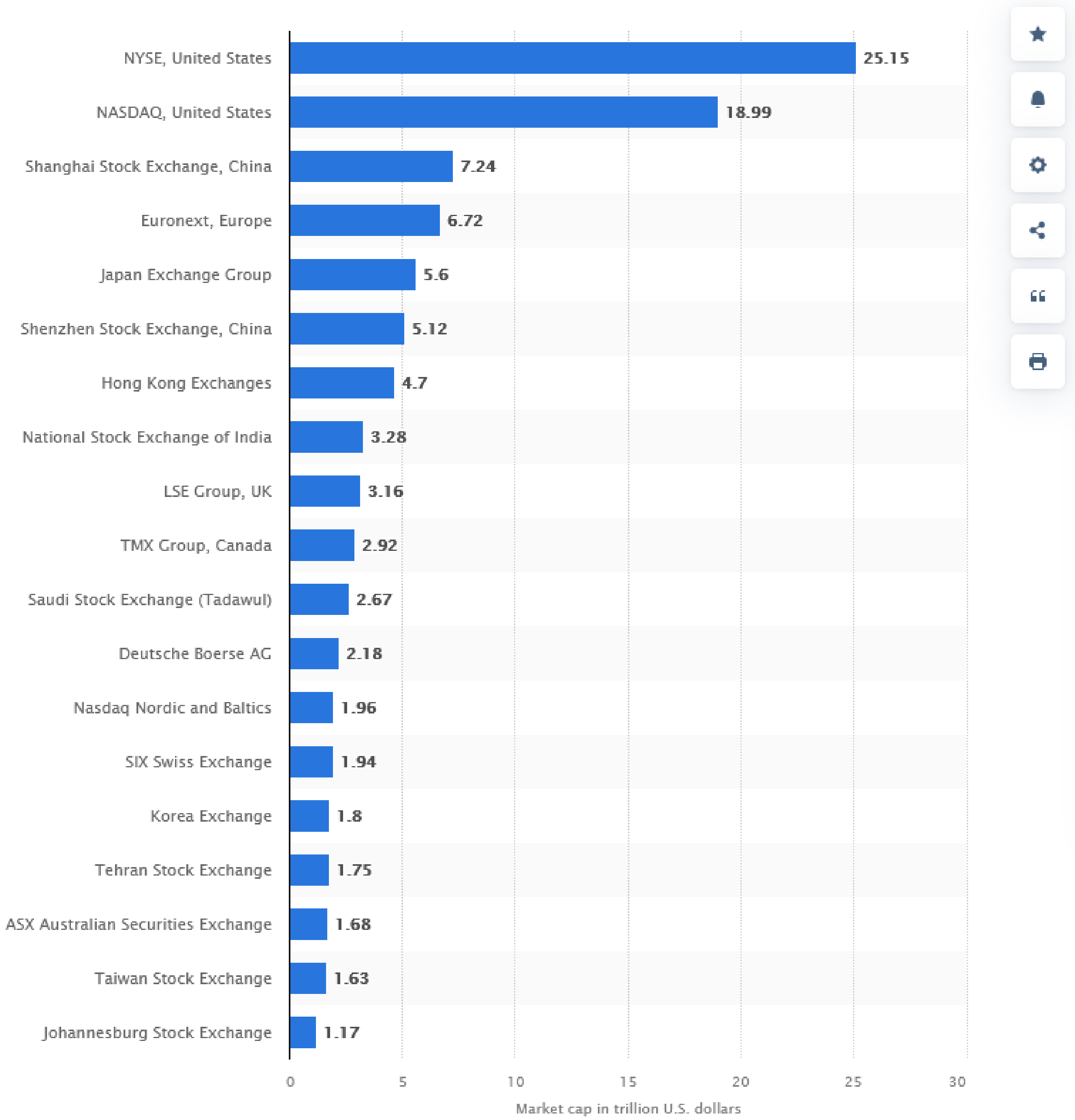

Fig.2. Stock exchange operators worldwide as of March 2023, by market capitalization of listed companies.

Stock markets are venues where buyers and sellers meet to exchange equity shares of public corporations. Stock markets are components of a free-market economy because they enable democratized access to investor trading and exchange of capital. Stock markets create efficient price discovery and efficient dealing. The stock market allows buyers and sellers of securities to meet, interact, and transact. The markets allow for price discovery for shares of corporations and serve as a barometer for the overall economy. Buyers and sellers are assured of a fair price, high degree of liquidity, and transparency as market participants compete in the open market.

The Functions of a Stock Market.

The stock market ensures price transparency, liquidity, price discovery, and fair dealings in trading activities.

The stock market guarantees all interested market participants have access to data for all buy and sell orders, thereby helping in the fair and transparent pricing of securities. The market also ensures efficient matching of appropriate buy and sell orders.

Stock markets need to support price discovery where the price of any stock is determined collectively by all of its buyers and sellers. Those qualified and willing to trade should get instant access to place orders and the market ensures that the orders are executed at a fair price.

Traders on the stock market include market makers, investors, traders, speculators, and hedgers. An investor may buy stocks and hold them for the long term, while a trader may enter and exit a position within seconds. A market maker provides necessary liquidity in the market, while a hedger may trade in derivatives.

- New York Stock Exchange (NYSE):

The NYSE is the world's largest stock exchange, offering icons and entrepreneurs the opportunity to raise capital and change the world.

The NYSE is owned by Intercontinental Exchange, an American holding company that it also lists (NYSE: ICE). Previously, it was part of NYSE Euronext (NYX), which was formed by the NYSE's 2007 merger with Euronext.

According to a Gallup poll conducted in 2022, approximately 58% of American adults reported having money invested in the stock market, either through individual stocks, mutual funds, or retirement accounts.

- Top NYSE Stocks by Dollar Value:

- Exxon Mobil Corp. (XOM) $110,500.5.
- Citigroup, Inc. (C) $77,714.8.
- Pfizer, Inc. (PFE) $76,547.1.
- International Business Machines (IBM) $60,407.1.

Note: **The Selection of company:**

We have collected data for (International Business Machines company) IBM stock. This is provided by New York Stock Exchange (NYSE) which is the world's largest stock exchange.

1.1. Project Motivations:

Stock price prediction is an important problem. With a successful model for stock prediction, we can gain insight about market behavior over time, spotting trends that would otherwise not have been noticed. With the increasingly computational power of the computer, Deep Learning (DL) Specifically, Recurrent Neural Network (RNN) will be an efficient method to solve this problem.

In this project, we will introduce a framework in which we integrate user predictions into the current Deep Learning (DL) algorithm using public historical data to improve our results. The motivated idea is that, if we know all information about todays stock trading (of all specific traders), the price is predictable. Thus, if we can obtain just a partial information, we can expect to improve the current prediction. our motivation is to design strong Deep Learning model that will benefit everyone and helps investors achieve the highest possible profit return and building an easy and secure trading platform.

1.2. Project Aims:

Stock Price Prediction is the task of forecasting future stock prices based on historical data and various market indicators. It involves using statistical models and machine learning algorithms to analyze financial data and make predictions about the future performance of a stock. The goal of stock price prediction is to help investors make informed investment decisions by providing a forecast of future stock prices.

The entire idea of predicting stock prices is to gain significant profits. Predicting how the stock market will perform is a hard task to do. There are other factors involved in the prediction, such as physical and psychological factors, rational and irrational behavior, and so on. All these factors combine to make share prices dynamic and volatile. This makes it very difficult to predict stock prices with high accuracy.

1.3. Project Contributions

The contribution of this Project is as follows:

(i) We present an overview of all the key technical approaches employed for the prediction of stock prices.

(ii) We explored various challenges in each approach, along with the future scope in each research and drew a comparative study of these approaches.

(iii) First, stock technical indicators are considered to identify the uptrend in stock prices. We consider moving average technical indicators: moving average 30 days, moving average 50 days, and moving average 24 hours.

(iv) We propose a hybrid stock prediction model using the LSTM and BI-GRU.

(v) We Design more than one prediction model, compared the results, and chose the best one.

(vi) helps investors achieve the highest possible profit return and building an easy and secure trading platform.

Chapter 2

Economically Feasibility Study And Engineering Standards.

Relation with the Environment and Economic Benefits

As the economy has grown rapidly in recent years, more and more people have begun putting their money into the stock market. Thus, predicting trends in the stock market is regarded as a crucial endeavor, and one that has proven to be more fruitful than others. Profitable investments will result in rising stock prices. Investors face significant difficulties making stock market-related predictions due to the lack of movement and noise in the data.

Stock Price forecasting is the task of forecasting future stock prices based on historical data and various market indicators. It involves using statistical models and machine learning algorithms to analyze financial data and make predictions about the future performance of a stock [4]. The goal of stock price prediction is to help investors make informed investment decisions by providing a forecast of future stock prices and achieving the highest possible profit return, which is beneficial to the global economy [5].

.

Engineering Standards

Standard for Operator Interfaces of Artificial Intelligence: **IEEE 2941-2021**

This standard defines a set of operator interfaces frequently found in artificial intelligence (AI) applications, where the AI operators refer to the standard building blocks and primitives for performing basic AI operations. It defines the functionality and the specific input and output operands of an AI operator. It takes into account both generality and efficiency, and covers various types of operators, such as those related to basic mathematics, neural network, and machine learning [6].

Chapter 3

Related Work.

Related Work

Many researchers have devoted time and energy to studying the topic of stock price prediction because of its significance [7,8]. In this section, the previous work efforts about stock market prediction will be presented. In [9], deep learning models for stock price prediction have been implemented. The proposed models are; Long-Short Term – Memory (LSTM) and Convolutional Neural Network (CNN) using the Hill Climbing (HC) heuristic. The proposed models have been used of predicting the closing price of 25 companies trading on the Bucharest Stock Exchange. Although the LSTM generates a larger total profit, the results show that the dual deep net contributes to a unique aspect of that profit in the investing simulation. When comparing the study's 25 companies, CNN has a larger profit in terms of gained market share than lost market share. While the annualized return yield from HC-LSTM is higher, the sharpness ratio is higher thanks to the contributions of HC-CNN.

As provided in [10], the Adaptive Neuro-Fuzzy Inference System (ANFIS), SVM, and Artificial Bee Colony (ABC) have been introduced for predicting the stock price in the future. At first, in order to get the most accurate forecasts possible, technical indicators are optimized using ABC. Next, ANFIS has been used to forecast long-term price changes in the stocks. Finally, SVM was used to establish a link between the stock price and a technical indicator in order to reduce the forecasting errors of the presented model. Computational results have shown the efficacy of ABC-ANFIS-SVM as a tool for stock price forecasting, which will aid traders and investors in identifying stock price trends and represent a breakthrough in algorithmic trading. As illustrated in [11], a Deep Learning Stock Market Prediction (DLSMP) has been proposed. The main aim of the proposed method was focused on stock market group predictions. To achieve such aim, Decision Tree, Bagging, Random Forest, Adaptive boosting (Adaboost), gradient boosting, and eXtreme gradient boosting (XGBoost), and Artificial Neural Networks (ANN), Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) were employed based on historical record of ten years of data. Experimental results in [11] have showed that LSTM is more accurate.

As introduced in [12], an Innovative Neural Network Approach (INNA) has been developed to give more accurate stock market prediction. The proposed approach uses deep long short-term memory with embedded layer. Also, in [13], a Stock Prediction System (SPS) has been proposed. The proposed SPS involves; 1) representing numerical price data by technical indicators through technical analysis and representing textual news articles by sentiment vectors through sentiment analysis, 2) setting up a layered deep learning model to learn the sequential information within market snapshot series that is constructed by the technical indicators and news sentiments, and 3) setting up a fully connected neural network to make stock predictions. The findings obtained show that for both the validation and test sets, the proposed method outperforms the baselines across both metrics.

As presented in [14], a new method has been proposed to predict stock market closing price in the next day. The proposed method called Convolutional Neural Networks, Bi-Directional Long Short-Term Memory, and Attention Mechanism (CNN-BiLSTM-AM). CNN-BiLSTM-AM splits into three stages which are; CNN was employed to extract features from input data. Then, the extracted feature data was learned and predicted using BiLSTM. Finally, the impact of the feature states of the time series data at various points on the prediction outcomes can be captured using AM. The results obtained demonstrated that the proposed CNN-BiLSTM-AM performs well where R^2 is the closest to 1. Also, authors in [15] use several machine learning models to predict stock market movements through Spark MLlib using PySpark. These models are; linear regression, decision tree, random forest, and generalized linear regression. The results obtain demonstrate that generalized linear regression is more accurate model.

Predicting stock market close prices using ANNs has been done previously in [16]. There are a number of limitations inherent to standalone ANNs that lead to poorer prediction accuracy. The utilization of hybrid models allows us to get around this limitation. Stock market forecasting using a combination of artificial intelligence networks and particle swarm optimization has been reported in the literature [17]. This article proposes a new improved method called Particle Swarm Optimization with Center of Gravity (PSOCoG) for rapidly achieving high-accuracy prediction. As a result, when comparing performance on the S&P 500 dataset, ANN-PSOCoG was 13% more accurate in its predictions than ANN-Standard PSO, 17% more accurate than SPSOCOG, nearly 20% more accurate than SPSO, and nearly 25% more accurate than ANN. When applying all of these methods to the DJIA dataset, ANN-PSOCoG outperformed ANN-SPSO by about 18%, SPSOCOG by about 24%, SPSO by about 33%, and ANN by about 42%. In addition, the proposed framework is examined in a COVID-19 infection model. For the S&P 500, GOLD, NASDAQ-100, and CANUSD datasets, the results showed that the proposed model accurately predicted the closing price when MAPE, MAE, and RE values were extremely small.

As presented in [18], a novel model which is called Improved Particle Swarm Optimization (IPSO) and Long-Short Term Memory (LSTM) has been proposed to forecast stock price. Actually, IPSO was used to set the hyperparameter of LSTM. As demonstrated experimentally in [18] the proposed model outperformed support-vector regression, LSTM, and PSO-LSTM on the Australian stock market index. These findings demonstrated that the proposed model is highly trustworthy and has excellent predictive ability. The authors in [19] used 1D DenseNet and an autoencoder to forecast the price of stocks at the end of the day. The computed stock technical indicators (STIs) were first fed into an autoencoder for dimensionality reduction, which led to lower levels of correlation between them. The 1D DenseNet was fed both the STIs and the Yahoo finance data. The 1D DenseNet's output features were fed into the 1D DenseNet's softmax layer for closing stock price prediction across time horizons.

As illustrated in [20], a new forecasting method in the neural network structure based on the Induced Ordered Weighted Average (IOWA) Weighted Average (WA) and fuzzy time series. First, it makes a theoretical contribution by suggesting a new IOWAWA layer in neural networks to deal with complex nonlinear prediction for a large data set. The second originality is using the method to forecast nonlinear financial market data. The proposed IOWAWA was compared to sixteen existing method and the results demonstrate that IOWAWA performs well.

Chapter 4

Proposed System

Proposed System

In this project we attempt to implement machine learning and Deep Learning approach to predict stock prices. Deep Learning is effectively implemented in forecasting stock prices. The objective is to predict the stock prices in order to make more informed and accurate investment decisions. We propose a stock price prediction system that integrates mathematical functions, Deep Learning, and other external factors for the purpose of achieving better stock prediction accuracy and issuing profitable trades.

our motivation is to design strong Deep Learning model that will benefit everyone and helps investors achieve the highest possible profit return and building an easy and secure trading platform.

4.1. Project Aims:

Stock Price Prediction is the task of forecasting future stock prices based on historical data and various market indicators. It involves using statistical models and machine learning algorithms to analyze financial data and make predictions about the future performance of a stock. The goal of stock price prediction is to help investors make informed investment decisions by providing a forecast of future stock prices.

The entire idea of predicting stock prices is to gain significant profits. Predicting how the stock market will perform is a hard task to do. There are other factors involved in the prediction, such as physical and psychological factors, rational and irrational behavior, and so on. All these factors combine to make share prices dynamic and volatile. This makes it very difficult to predict stock prices with high accuracy.

4.2. Project Methodology:

A project system is employed for the suggested solution, with a focus on scalability and sustainability. An overall outline of the proposed system is shown in Fig. 3.

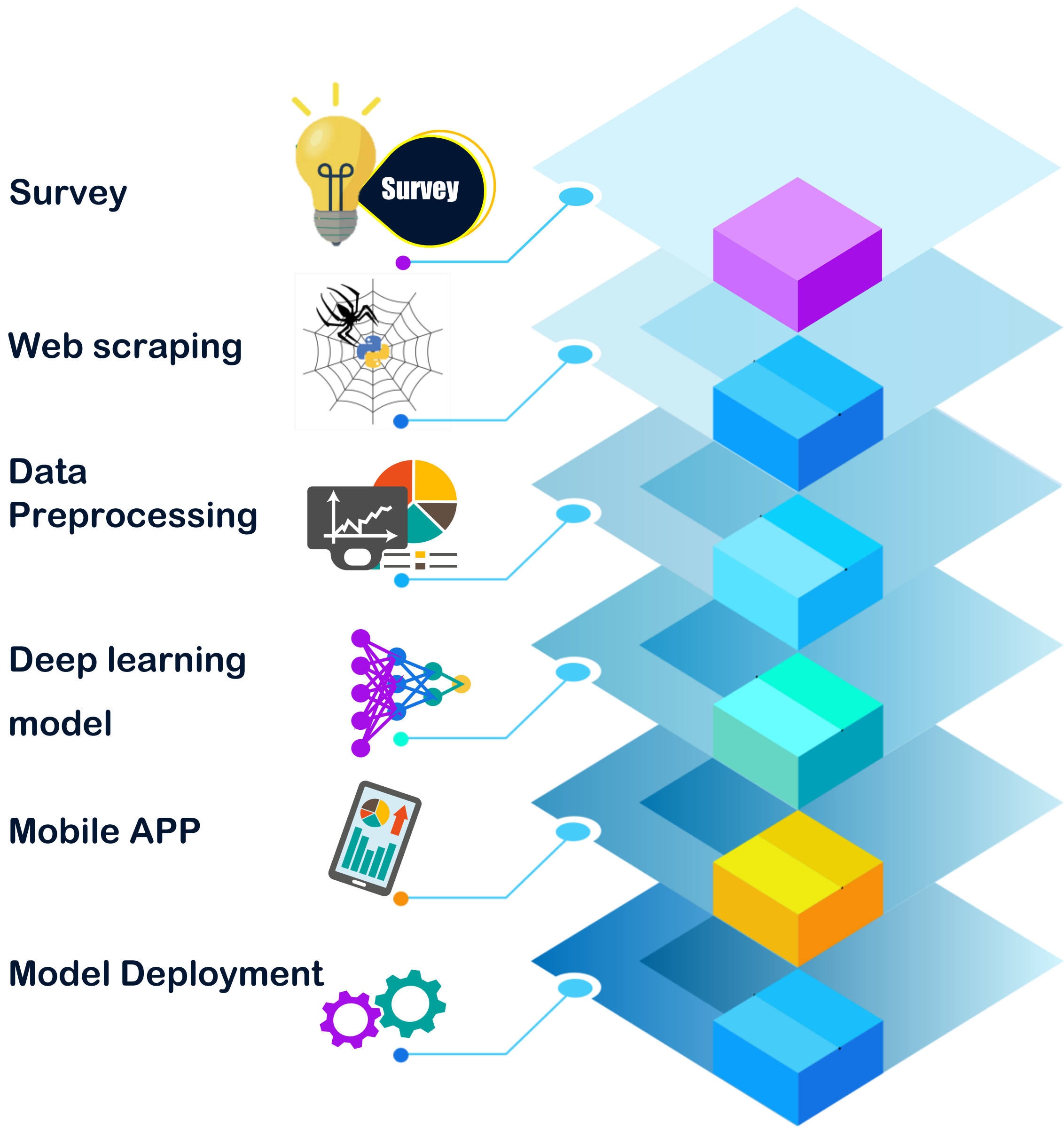

Fig.3. Project Methodology

4.2.1. Survey.

Pre-building system stage. At this stage, we conducted a survey on the global Robinhood platform for stock trading to find out the opinions of investors whether they are interested about using artificial intelligence in trading or not. Now we will clarify the question we asked and some of the investors' responses as shown through fig. 4 to fig.7.

Fig.4. survey question.

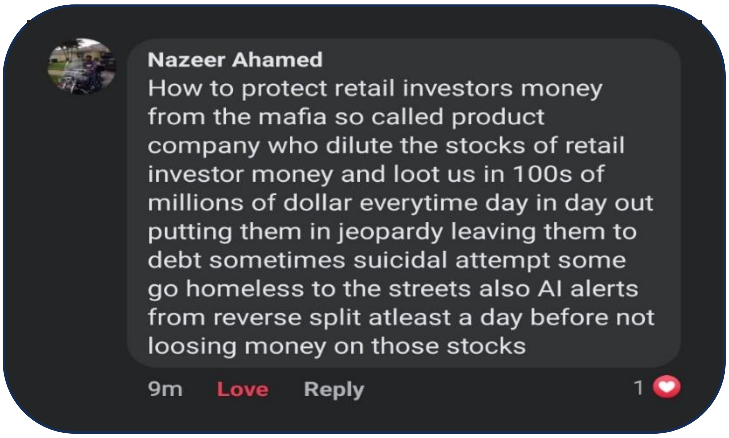

Fig.5. Answer survey question

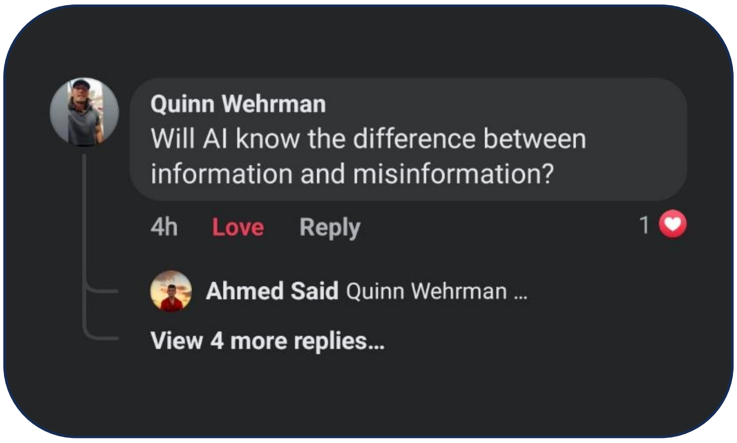

Fig.6. Answer survey question

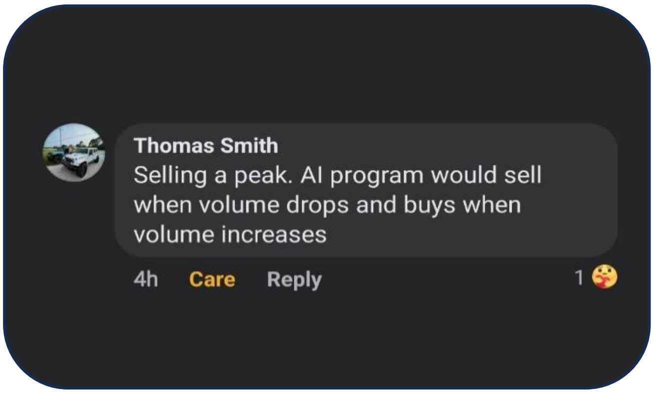

Fig.7. Answer survey Question

4.2.2. Web scraping.

Phase-I

- **Web Definition**

  World Wide Web (WWW), commonly known as the Web, is an information system enabling documents and other web resources to be accessed over the Internet

- **Data Shape**

  Factual information (such as measurements or statistics) used as a basis for reasoning, discussion, or calculation.

  In general Data can come in the form of text, observations, figures, images, numbers, graphs, or symbols. For example, data might include individual prices, weights, addresses, ages, names, temperatures, dates, or distances. Data is a raw form of knowledge.

  In our case We will extract data from Financial Markets and analysis consumer trends and understanding which direction the company should move in the future.

Phase II

- **Web Scraping Definition**

  "Web scraping" also called crawling or spidering, is the automated gathering of data from an online source usually from a website [7].

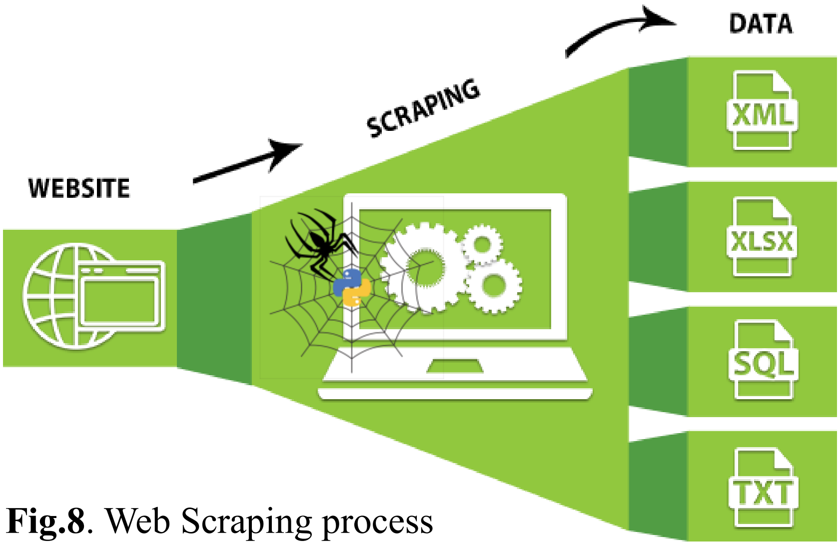

Fig.8. Web Scraping process

- **How Web Scrapers Work.**
    - Web Scrapers can extract all the data on particular sites or the specific data that a user wants. Ideally, it's best if you specify the data, you want so that the web scraper only extracts that data quickly. For example, you might want to scrape an Amazon page for the types of juicers available, but you might only want the data about the models of different juicers and not the customer reviews.

    - So, when a web scraper needs to scrape a site, first the URLs are provided. Then it loads all the HTML code for those sites and a more advanced scraper might even extract all the CSS and Java script elements as well. Then the scraper obtains the required data from this HTML code and outputs this data in the format specified by the user. Mostly, this is in the form of an Excel spreadsheet or a CSV file, but the data can also be saved in other formats, such as a JSON file.
- **Price Monitoring.**
    - We built a script to track multiple stock prices, organize them into an easy-to-read CSV file that will update itself with the push of a button, and collect hundreds of data points in a few seconds.
    -
- **Why is Python a popular programming language for Web Scraping.**

  Python seems to be in fashion these days! It is the most popular language for web scraping as it can handle most of the processes easily. It also has a variety of libraries that were created specifically for Web Scraping.

    - **Scrapy** is a very popular open-source web crawling framework that is written in Python. It is ideal for web scraping as well as extracting data using APIs.

    - **Beautiful soup** is another Python library that is highly suitable for Web Scraping. It creates a parse tree that can be used to extract data from HTML on a website. Beautiful soup also has multiple features for navigation, searching, and modifying these parse trees

Fig.9. BeautifulSoup Library

Libraries we used:

- BeautifulSoup v4.11.1
- yfinance v0.1.85
- datetime

4.2.3. Data Preprocessing.

- The definition of Data Preprocessing.

- Data preprocessing, a component of data preparation, describes any type of processing performed on raw data to prepare it for another data processing procedure [22,23]. It has traditionally been an important preliminary step for the data mining process. More recently, data preprocessing techniques have been adapted for training machine learning models and AI models and for running inferences against them.

- Data preprocessing transforms the data into a format that is more easily and effectively processed in data mining, machine learning and other data science tasks. The techniques are generally used at the earliest stages of the machine learning and AI development pipeline to ensure accurate results [24,25].

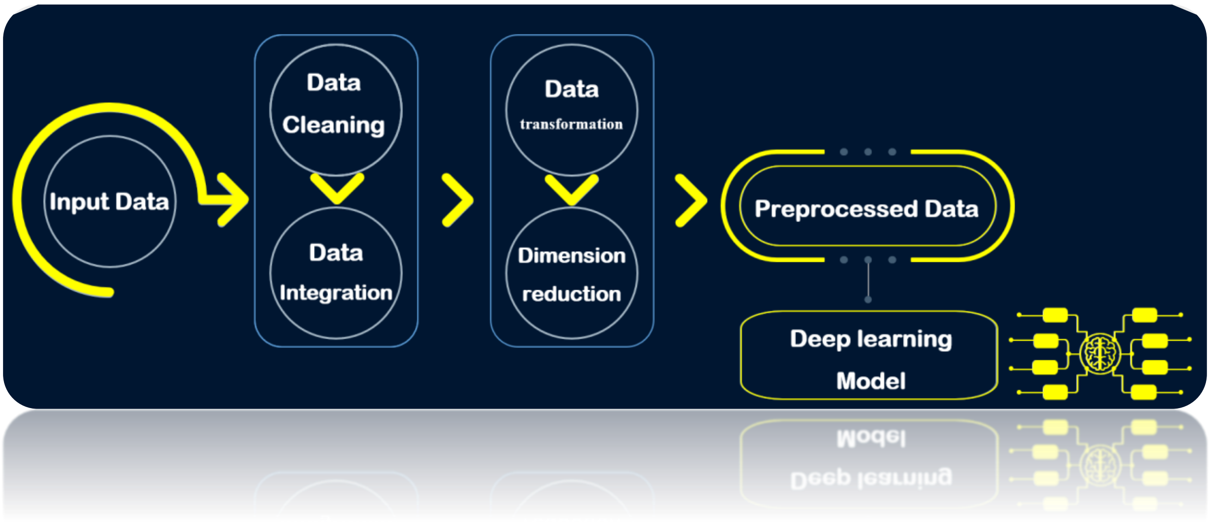

Fig.10. Data Preprocessing Process.

- **Data Cleaning definition.**
    - the process of fixing or removing incorrect, corrupted, incorrectly formatted, duplicate, or incomplete data within a dataset.

Fig.11. Data Cleaning Cycle

- **Data integrity.**
    - Data integrity refers to the accuracy and consistency (validity) of data over its lifecycle. Compromised data, after all, is of little use to enterprises, not to mention the dangers presented by sensitive data loss. For this reason, maintaining data integrity is a core focus of many enterprise security solutions.

    - Data integrity can be compromised in several ways. Each time data is replicated or transferred, it should remain intact and unaltered between updates. Error checking methods and validation procedures are typically relied on to ensure the integrity of data that is transferred or reproduced without the intention of alteration.
- **Data transformation.**
    - Here, data scientists think about how different aspects of the data need to be organized to make the most sense for the goal. This could include things like structuring unstructured data, combining salient variables when it makes sense or identifying important ranges to focus on.
- **Data reduction.**
    - Raw data sets often include redundant data that arise from characterizing phenomena in different ways or data that is not relevant to a particular ML, AI or analytics task. Data reduction uses techniques like principal component analysis to transform the raw data into a simpler form suitable for particular use cases [26,27].

    - Data reduction. Data scientists often need to combine a variety of data sources to create a new AI or analytics model. Some of the variables may not be correlated with a given outcome and can be safely discarded. Other variables might be relevant, but only in terms of relationship -- such as the ratio of debt to credit in the case of a model predicting the likelihood of a loan repayment; they may be combined into a single variable. Techniques like principal component analysis play a key role in reducing the number of dimensions in the training data set into a more efficient representation.

- **Feature Engineering.**

Feature engineering, as noted, involves techniques used by data scientists to organize the data in ways that make it more efficient to train data models and run inferences against them [29,30]. These techniques include the following:

- **Feature scaling or normalization.** Often, multiple variables change over different scales, or one will change linearly while another will change exponentially. For example, salary might be measured in thousands of dollars, while age is represented in double digits. Scaling helps to transform the data in a way that makes it easier for algorithms to tease apart a meaningful relationship between variables.

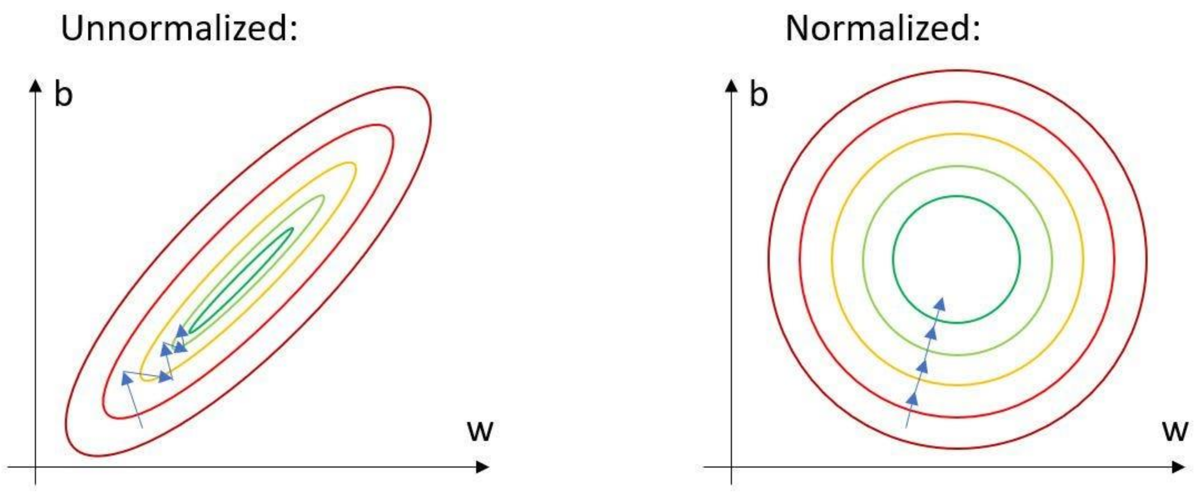

Fig.12. Data Normalization

- **The Description of dataset:**
    - **The Selection of company:** We have collected data for (International Business Machines company) IBM stock. This is provided by New York Stock Exchange (NYSE) which is the world's largest stock exchange
    - data was extracted for period 3/25/2022 to 4/6/2020 interval to minutes and data shape (216883, 6).

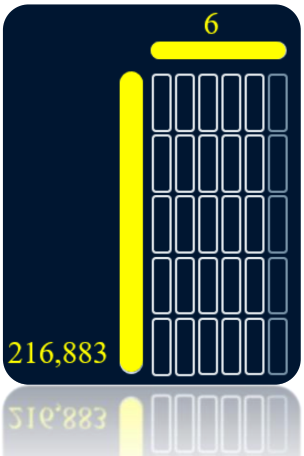

Fig.13. Data Shape.

- **Adjustment of closing price:**
  The last quoted price for a stock during a day is called its closing price. This is the standard price considered during financial time series analysis. This, however, needs to be adjusted for corporate actions before doing any analysis on historical data. These actions could be stock splits, dividends or rights offerings. All of these affect the "nominal" price of the stock though none of them affects its "true" price. For proper training of the LSTM and GRU, the price effect of these actions needs to be removed from the nominal stock price. The closing price after these adjustments is called the adjusted close price and has been used in our analysis.

4.2.4. Machine Learning model (ML model):

we discuss some of the most common traditional machine learning algorithms that are used for machine vision. With so-called supervised machine learning, we can model relationships between the target prediction output and the input features. To achieve good performance, the input features must be carefully selected by people before the algorithm is trained on the data [31,32]

Today, traditional machine learning algorithms are significantly overshadowed by deep learning. However, they are still well suited for many applications independently or as a support in complex pipelines. Traditional machine learning is able to perform two tasks: regression and classification.

- **Linear, polynomial and logistic regressions.**

  Linear regression is a model that assumes a linear relationship between the input variables (x) and the output variable (y). The main goal of a linear regression model is to fit a linear function between data points, i.e. to find the optimal values of intercept and coefficients, so that the error is minimized.

  So, how do we achieve the optimal linear relationship? Let's have a closer look.
  Let's say we have an input with a set of features {Feature_1, Feature_2, ... , Feature_N} and a mathematical model showing us how to make a prediction. In our case, this is a linear function with a set of unknown coefficients {bias, C1, C2, ..., CN}

  $$\text{Prediction} = \text{bias} + C_1 \cdot \text{Feature}_1 + C_2 \cdot \text{Feature}_2 + \dots + C_N \cdot \text{Feature}_N$$

  To start our search for the optimal linear relationship, we can set the coefficients to random or at least intuitively reasonable values. Based on these, our model can make its first prediction. Now we can measure how far the predicted value is from the ground truth by computing the mean squared error:

  $$\text{Error} = 1/M\ [\ (\text{Prediction}_1 - \text{True Value}_1)^2 + \dots + (\text{Prediction}_M - \text{True Value}_M)^2]$$

  We want our predictions to be as close as possible to our ground truth values. To achieve this, we are going to update our coefficients in iterations by means of an optimization algorithm.

This way, we reduce the error with each iteration. Importantly, we want our model to generalize well on data the model has never seen before. For this purpose, we split our dataset into a training and a validation subset. We use the training dataset to adjust the coefficients and the validation subset to independently estimate how the model performs on unfamiliar data. The model's performance on the independent validation set is used for the final model selection.

In many cases, the data will be nonlinear in nature, requiring a nonlinear function to model. In that case, we can use polynomials When the number of input features is high, linear and polynomial regression models tend to overfit the data (meaning, they generalize poorly on data they have never seen). In that case, we can use other regression models, such as Ridge regression or Lasso regression, that include the regularization terms to reduce overfitting.

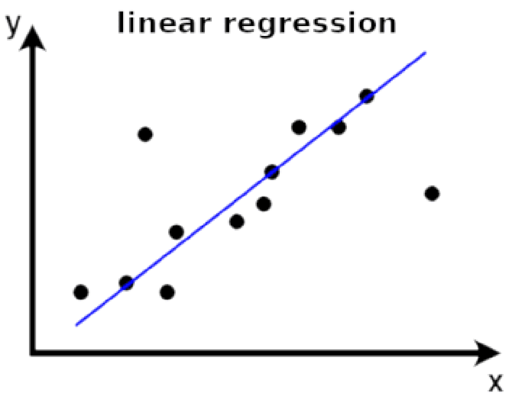

Fig.14. Linear Regression.

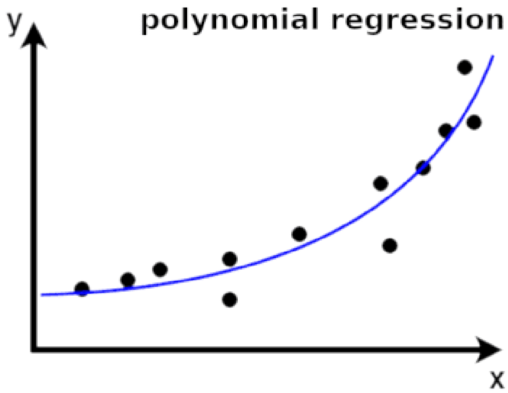

Fig.15. Polynomial Regression.

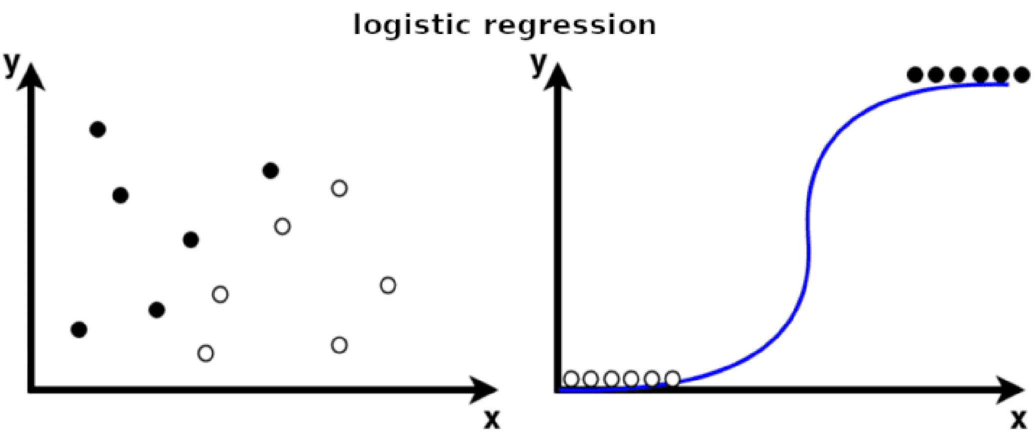

Fig.16. Logistic Regression.

- Overall Architecture:

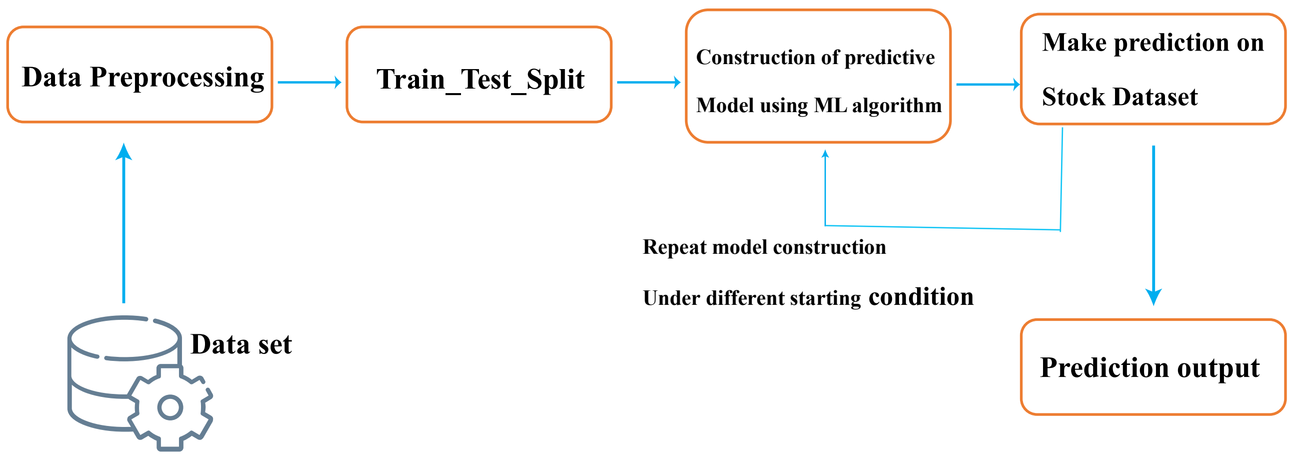

Fig.17. ML Model Architecture.

Results:

- Since the model's predictions were accurate enough, it failed to predict stock prices correctly, which prompted us to use a more complex and highly efficient model using Deep Learning.

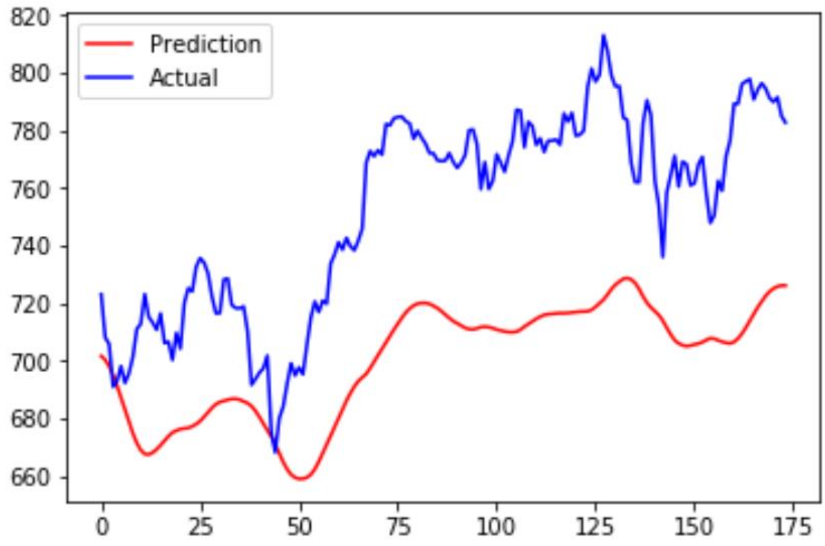

Fig.18. ML Model Results.

- **Libraries:**
    1. **Pandas 1.4.2**
    2. **Matplotlib 3.5.1**
    3. **Seaborn 0.11.2**
    4. **Sklearn 1.0.2**

4.2.5. Deep learning.

- **Deep learning Definition** is a subset of machine learning, which is essentially a neural network with three or more layers. These neural networks attempt to simulate the behavior of the human brain—albeit far from matching its ability—allowing it to "learn" from large amounts of data. While a neural network with a single layer can still make approximate predictions, additional hidden layers can help to optimize and refine for accuracy.
- **Deep Learning methodology.**

  Deep learning neural networks (DNN), or artificial neural networks (ANN), attempts to mimic the human brain through a combination of data inputs, weights, and bias. These elements work together to accurately recognize, classify, and describe objects within the data.

- **Deep neural networks** consist of multiple layers of interconnected nodes, each building upon the previous layer to refine and optimize the prediction or categorization. This progression of computations through the network is called forward propagation. The input and output layers of a deep neural network are called "visible layers". The input layer is where the deep learning model ingests the data for processing, and the output layer is where the final prediction or classification is made.

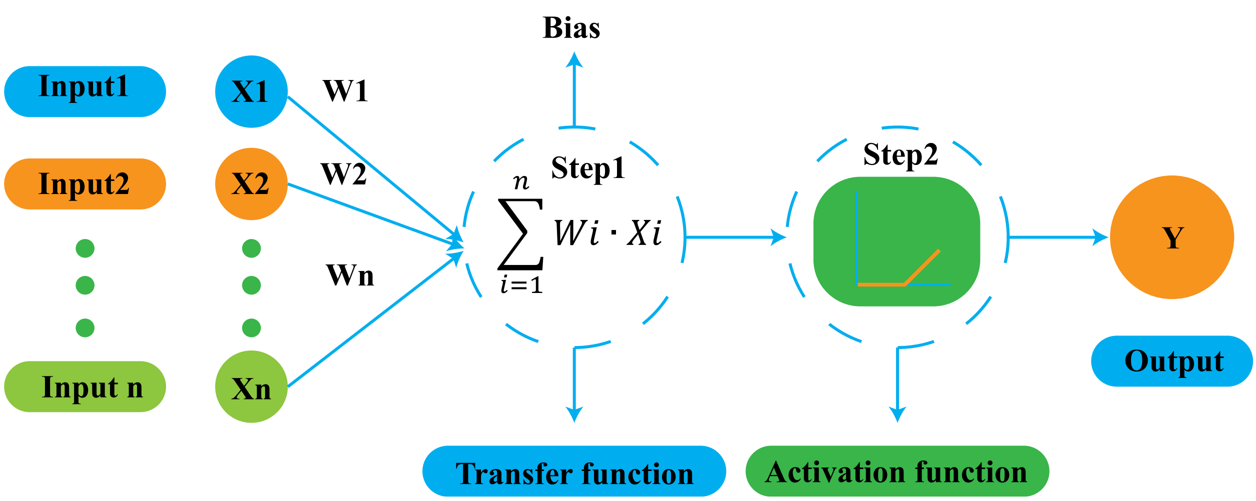

Fig.19. Simple Neural Network.

Another process called backpropagation uses algorithms, like gradient descent, to calculate errors in predictions and then adjusts the weights and biases of the function by moving backwards through the layers in an effort to train the model. Together, forward propagation and backpropagation allow a neural network to make predictions and correct for any errors accordingly. Over time, the algorithm becomes gradually more accurate.

- **The most popular two types of neural networks.**
    - **Convolutional neural networks (CNNs),** used primarily in computer vision and image classification applications, can detect features and patterns within an image, enabling tasks, like object detection or recognition [30]. In 2015, a CNN bested a human in an object recognition challenge for the first time [30,31].
    - **Recurrent neural networks (RNNs),** are typically used in natural language and speech recognition applications as it leverages sequential or times series data.

Note: We built a deep learning model in this project using the Recurrent Neural Network (RNN).

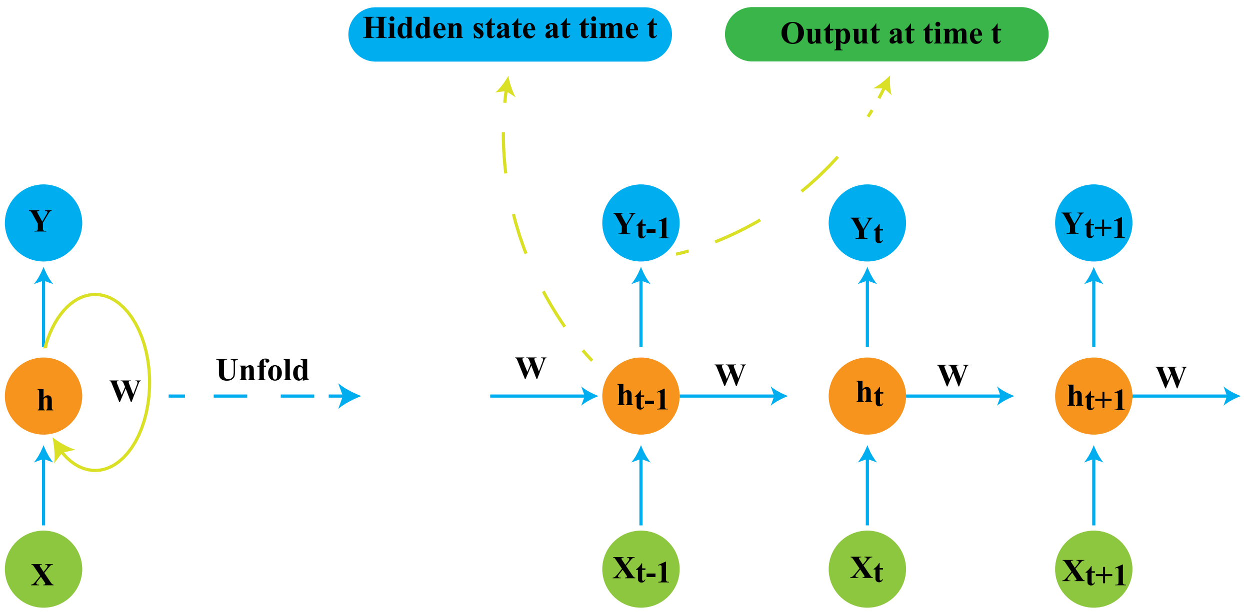

Fig.20. Recurrent Neural Network (RNN).

- **Deep learning applications.**

Real-world deep learning applications are a part of our daily lives, but in most cases, they are so well-integrated into products and services that users are unaware of the complex data processing that is taking place in the background. Some of these examples include the following:

- **Financial services:**
  Financial institutions regularly use predictive analytics to drive algorithmic trading of stocks, assess business risks for loan approvals, detect fraud, and help manage credit and investment portfolios for clients.

- **Customer service:**
  Many organizations incorporate deep learning technology into their customer service processes. Chatbots—used in a variety of applications, services, and customer service portals—are a straightforward form of AI. Traditional chatbots use natural language and even visual recognition, commonly found in call center-like menus. However, more sophisticated chatbot solutions attempt to determine, through learning, if there are multiple responses to ambiguous questions. Based on the responses it receives, the chatbot then tries to answer these questions directly or route the conversation to a human user.

- **Structure Chart.**

  A structure chart (SC) in software engineering and organizational theory is a chart which shows the breakdown of a system to its lowest manageable levels. They are used in structured programming to arrange program modules into a tree. Each module is represented by a box, which contains the module's name.

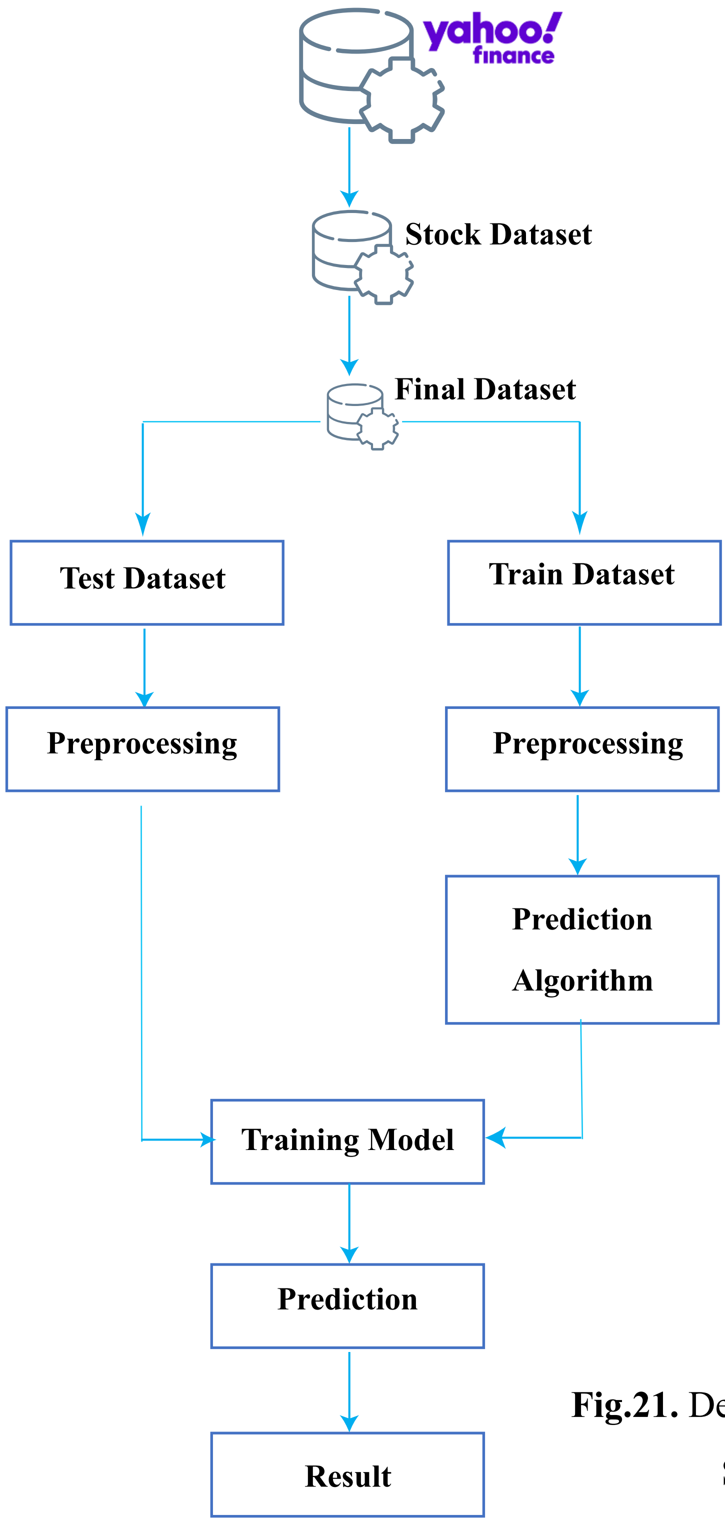

Fig.21. Deep Learning Model Structure.

- **Data Splitting:**

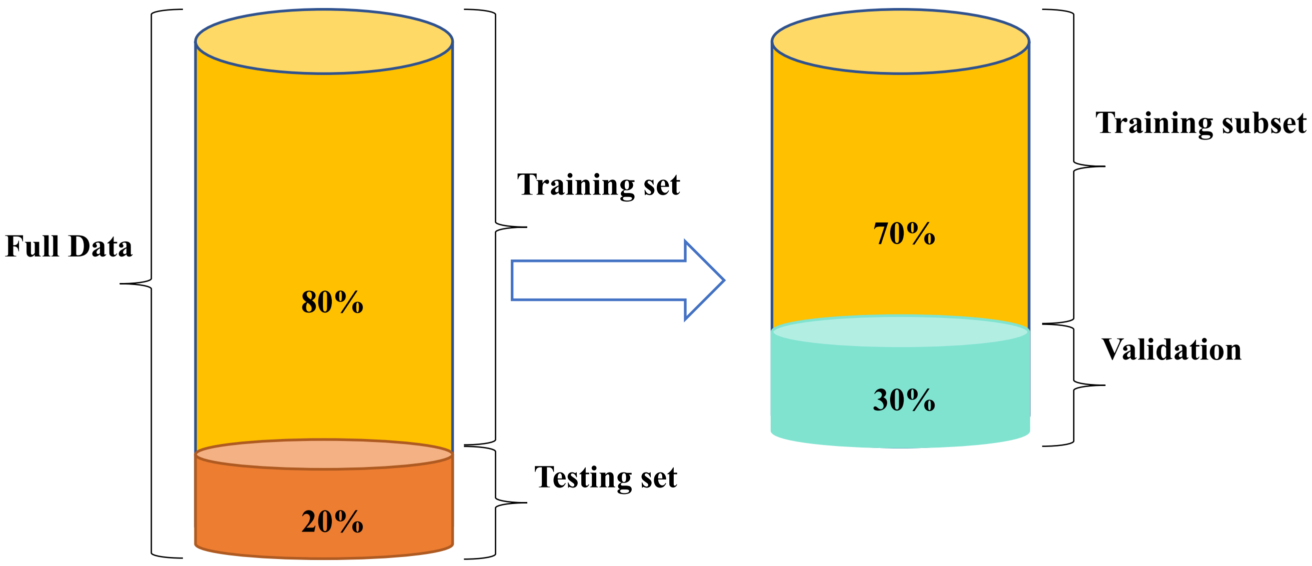

Fig.22. Data Splitting (Train, Test & Validation).

In this method the mostly large dataset is randomly divided to three subsets:

- **Training set is a subset of the dataset used to build our predictive model.**

- **Validation set is a subset of the dataset used to assess the performance of model built in the training phase. It provides a test platform for fine tuning model's parameters and selecting the best-performing model.**

- **Test set or unseen examples** is a subset of the dataset to assess the likely future performance of a model. If a model fit to the training set much better than it fits the test set, overfitting is probably the cause.

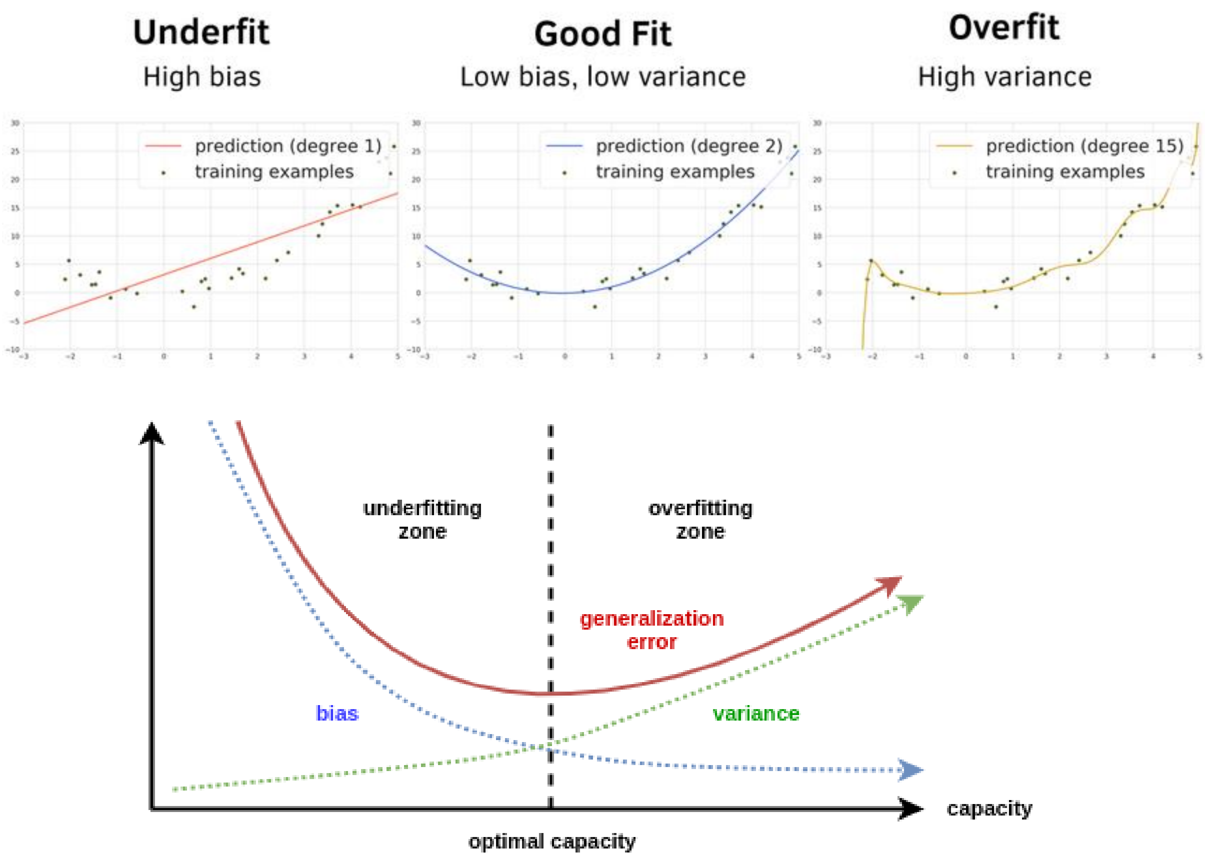

Fig.23. Overfitting, Good fitting & Underfitting.

- **Treatments:**
- **Regularization Techniques:**
    - **Lasso (L1).**
    - **Ridge (L2).**
    - **Elastic Net.**

- ### Regularization Techniques:

  Regularization is a technique used in machine learning and deep learning to prevent overfitting and improve the generalization performance of a model. It involves adding a penalty term to the loss function during training. This penalty discourages the model from becoming too complex or having large parameter values, which helps in controlling the model's ability to fit noise in the training data. Regularization methods include L1 and L2 regularization, dropout, early stopping.

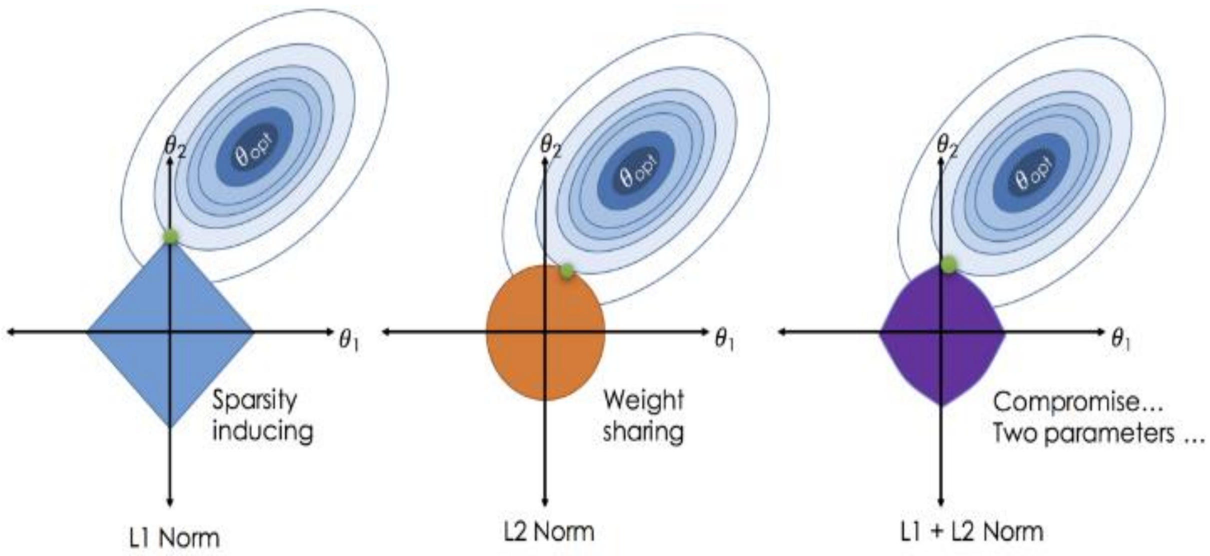

Fig.24. Regularization Techniques.

- **Vanishing Gradient Problem:**

  The vanishing gradient problem is caused by the fact that while the process of backpropagation goes on, the gradient of the early layers (the layers that are nearest to the input layer are derived by multiplying the gradients of the later layers (the layers that are near the output layer). Therefore, if the gradients of later layers are less than one, their multiplication vanishes at a particularly rapid pace.

- Backpropagation through the time.

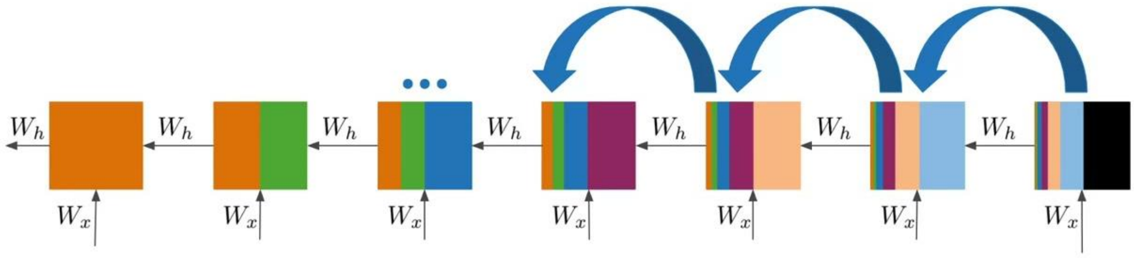

Fig.25. Vanishing Gradient Problem.

- **Overcoming vanishing gradient problem.**

Here are some methods that are proposed to overcome the vanishing gradient problem:

- Residual neural networks (ResNets)
- Multi-level hierarchy
- Long short term memory (LSTM)
- Faster hardware
- ReLU
- Batch normalization

- Long short-term memory (LSTM):

Long Short Term Memory (LSTM) was created specifically for the purpose of preventing the vanishing gradient problem. It manages to do that with the Constant Error Carousel (CEC). However, even in an LSTM, the gradients do tend to vanish; they just vanish at a far slower pace than they do in regular recurring neural networks.

Long Short-Term Memory Networks is a deep learning, sequential neural network that allows information to persist. It is a special type of Recurrent Neural Network which is capable of handling the vanishing gradient problem faced by RNN. LSTM was designed by Hochreiter and Schmidhuber that resolves the problem caused by traditional RNNs and machine learning algorithms. LSTM can be implemented in Python using the Keras library.

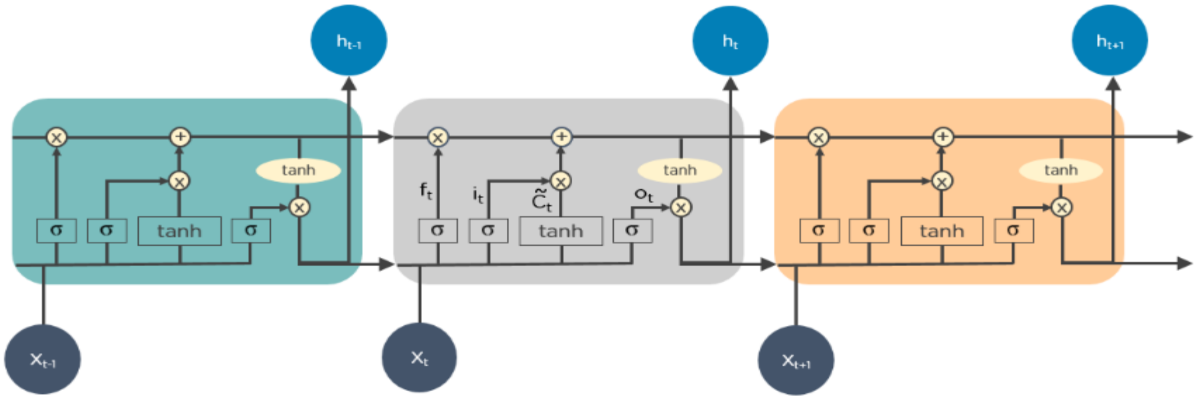

Fig.26. Long Short-Term Memory (LSTM).

- **LSTM Equations:**

$$
\begin{aligned}
f_t &= \sigma_g(W_f * x_t + U_f * h_{t-1} + V_f \odot c_{t-1} + b_f) \\
i_t &= \sigma_g(W_i * x_t + U_i * h_{t-1} + V_i \odot c_{t-1} + b_i) \\
c_t &= f_t \odot c_{t-1} + i_t \odot \sigma_c(W_c * x_t + U_c * h_{t-1} + b_c) \\
o_t &= \sigma_g(W_o * x_t + U_o * h_{t-1} + V_o \odot c_t + b_o) \\
h_t &= o_t \odot \sigma_h(c_t)
\end{aligned}
$$

- **Gate Recurrent Unit (GRU):**

GRU or Gate Recurrent Unit is an advancement of the standard Recurrent Neural Network (RNN).

GRUs are very similar to Long Short-Term Memory (LSTM). Just like LSTM, GRU uses gates to control the flow of information. They are relatively new as compared to LSTM. This is the reason they offer some improvement over LSTM and have simpler architecture.

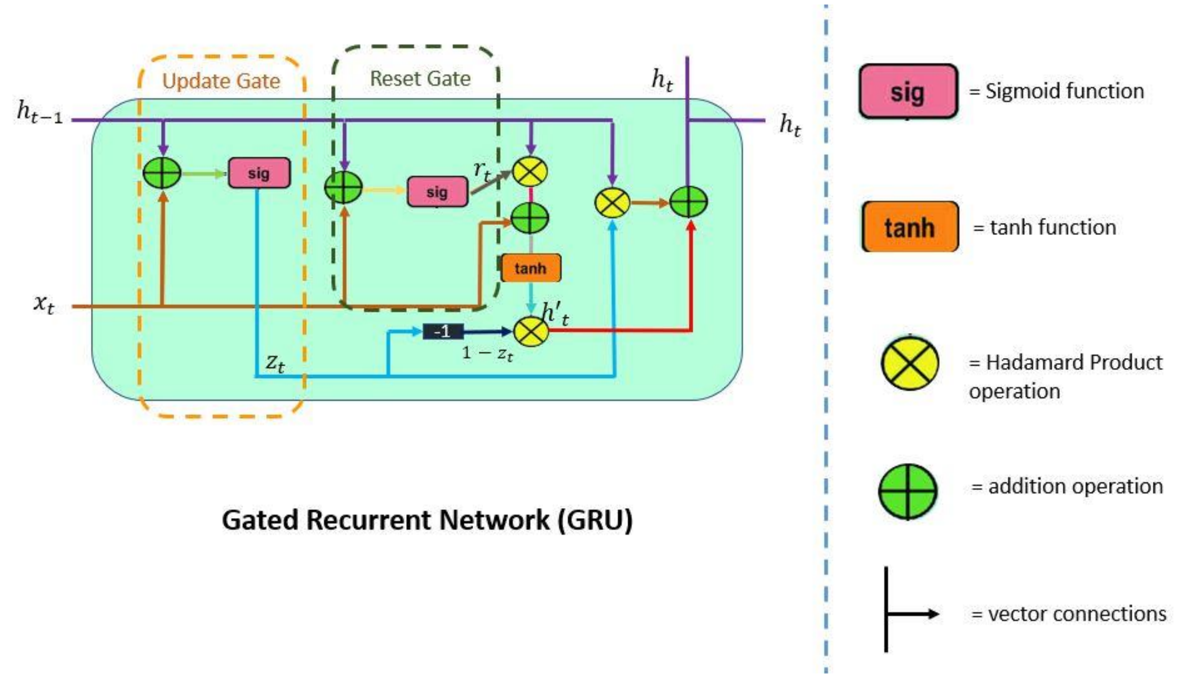

Fig.27. Gate Recurrent Unit (GRU).

- **GRU Equations:**

$$
\begin{aligned}
z_t &= \sigma_g(W_z x_t + U_z h_{t-1} + b_z) \\
r_t &= \sigma_g(W_r x_t + U_r h_{t-1} + b_r) \\
\hat{h}_t &= \phi_h(W_h x_t + U_h(r_t \odot h_{t-1}) + b_h) \\
h_t &= z_t \odot h_{t-1} + (1 - z_t) \odot \hat{h}_t
\end{aligned}
$$

- x_t: input vector
- h_t: output vector
- $\hat{h}_t$: candidate activation vector
- z_t: update gate vector
- r_t: reset gate vector
- W, U and b: parameter matrices and vector

- Bi-directional Gated Recurrent Unit (Bi-GRU):

A Bi-GRU layer works by applying two GRU layers on the data; one in the forward direction and one in the reverse direction.

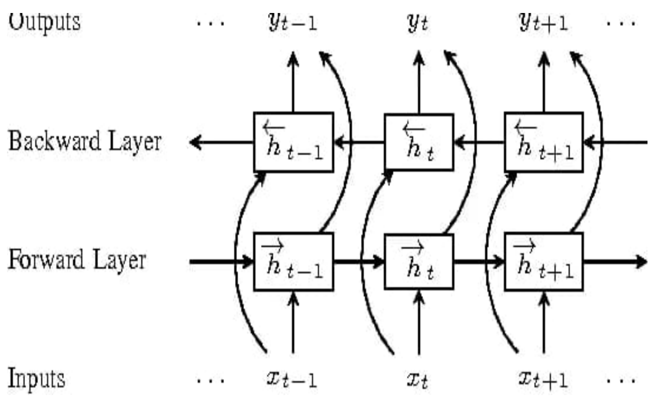

Fig.28. Bi-GRU

- **system configuration:**

This project can run on commodity hardware. We ran entire project on an Intel Core i7 processor with 8 GB Ram, 4 GB Nvidia Graphic Processor. First part of the is training phase which takes 50-60 minutes of time and the second part is testing part which only takes few seconds to make predictions and calculate accuracy.

- **Hardware Requirements:**
  - RAM: 4 GB
  - Storage: 500 GB
  - CPU: 2 GHz or faster
  - Architecture: 32-bit or 64-bit
- **Software requirements:**
  - Operating System: Windows 7 and above or Linux based OS or Mac-OS.
  - Python 3.5 in Jupyter-Notebook or Python 3.5 in Google Colab is used for data pre-processing, model training and prediction.

4.2.6. Mobile APP System?

1. UI Phase.
2. UX phase.

- **User Interface (UI).**

  User interface (UI) design is the process designers use to build interfaces in software or computerized devices, focusing on looks or style. Designers aim to create interfaces which users find easy to use and pleasurable.

- **Tools:**
  - **Adobe XD.**
  - **Adobe Illustrator.**

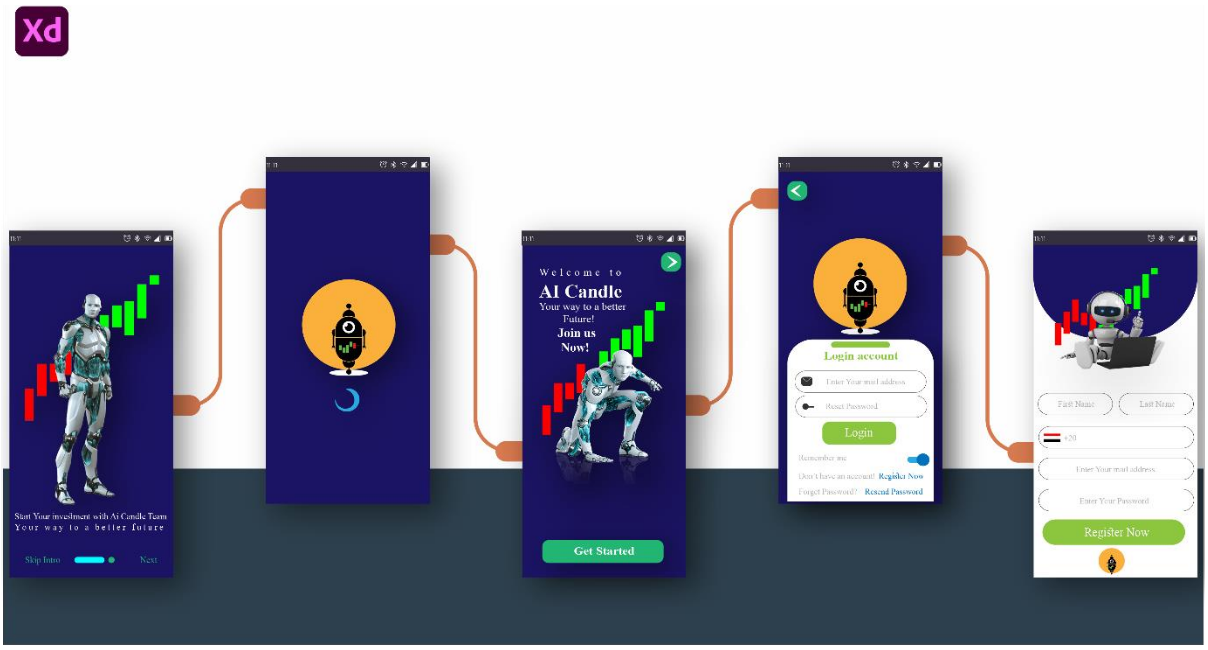

Fig.29. User Interface phase (Adobe XD).

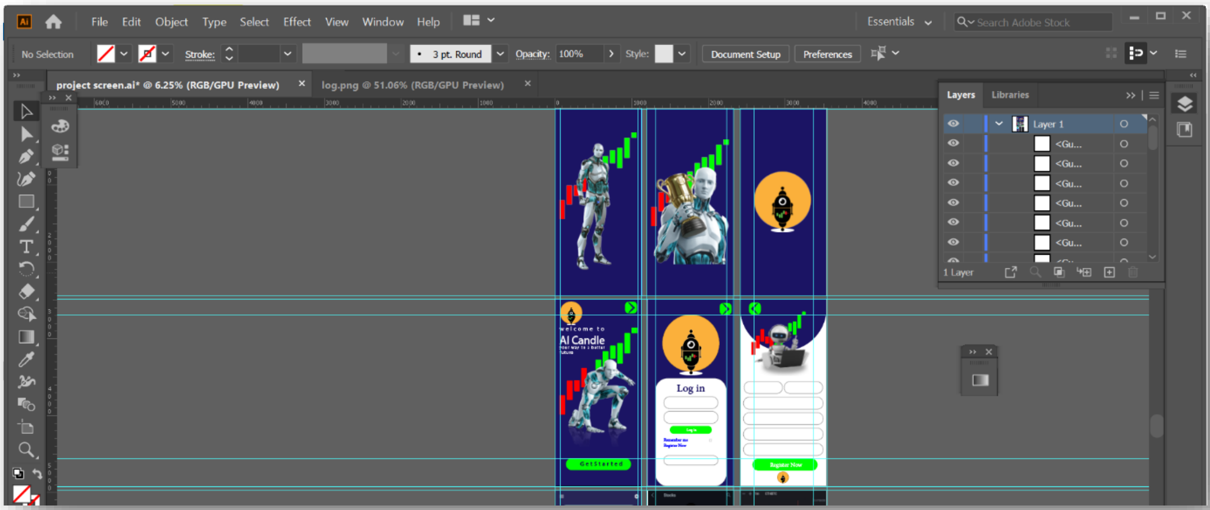

Fig.30. User Interface phase (Adobe Illustrator).

- **User Experience (UX).**

  User experience (UX) design is the process design teams use to create products that provide meaningful and relevant experiences to users. UX design involves the design of the entire process of acquiring and integrating the product, including aspects of branding, design, usability and function.

- Mobile APP Structure.

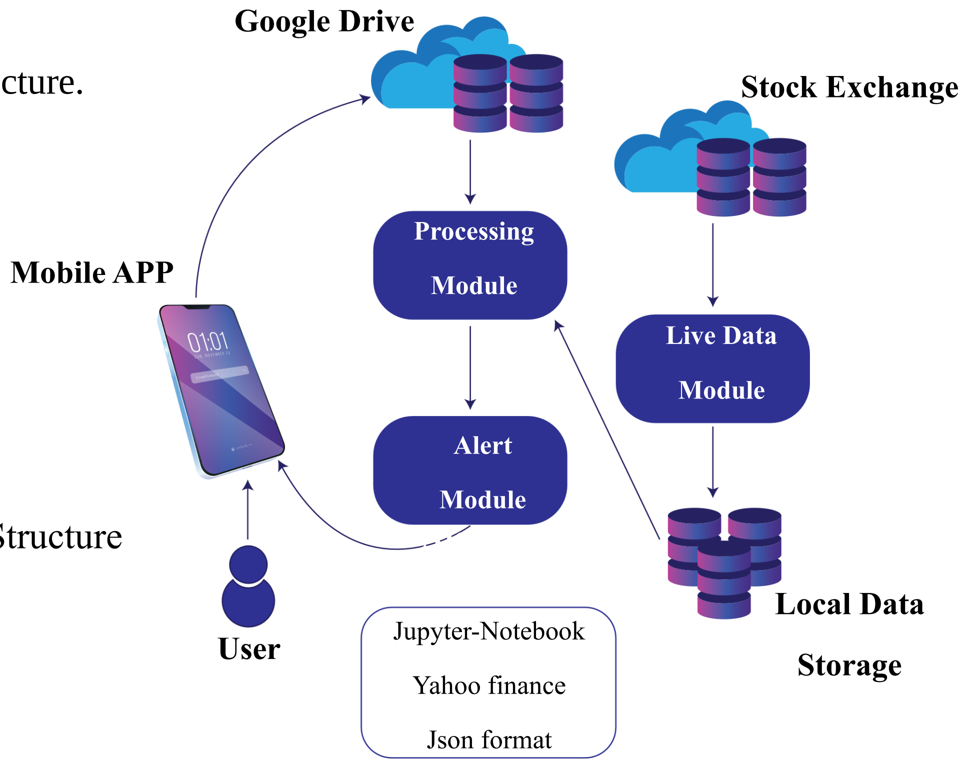

Fig.31. Mobile APP Structure

At this stage, we will build the mobile application using the flutter cross platform, & Dart programming language and then we will link it to our Deep Learning model so that this application serves as smart platform from which the investor will starting his safety journey in the world of trading.

- Flutter is a cross-platform framework that targets developing high-performance mobile applications.
- Flutter was publicly released in 2016 by Google.

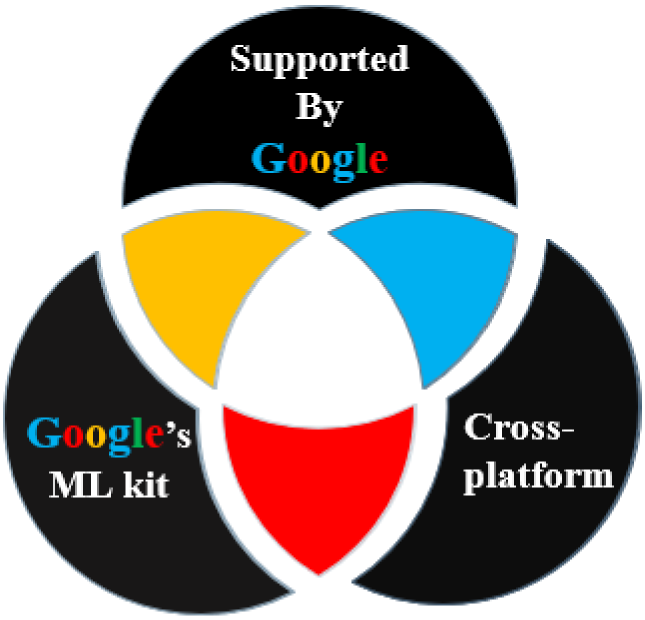

Fig.32. Flutter Features.

- **Traditional APP** Development

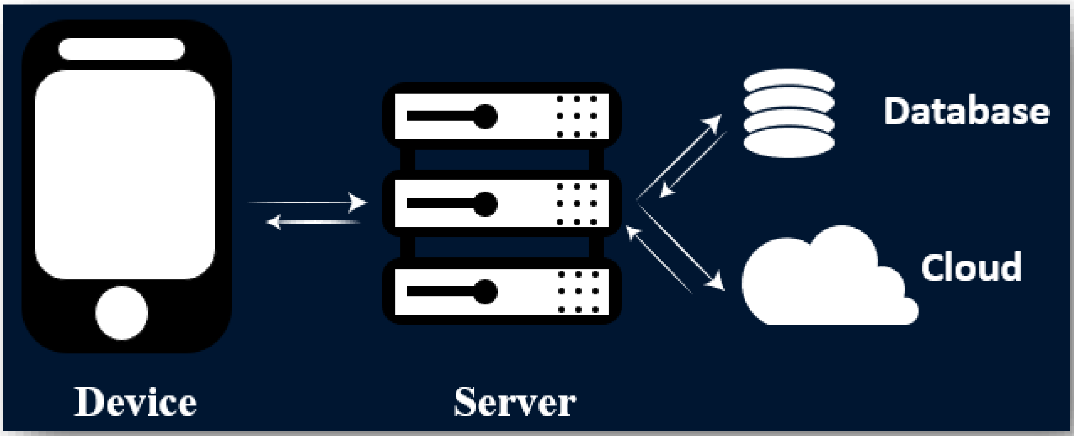

Fig.33. Traditional APP Development.

- **Firebase APP Development**

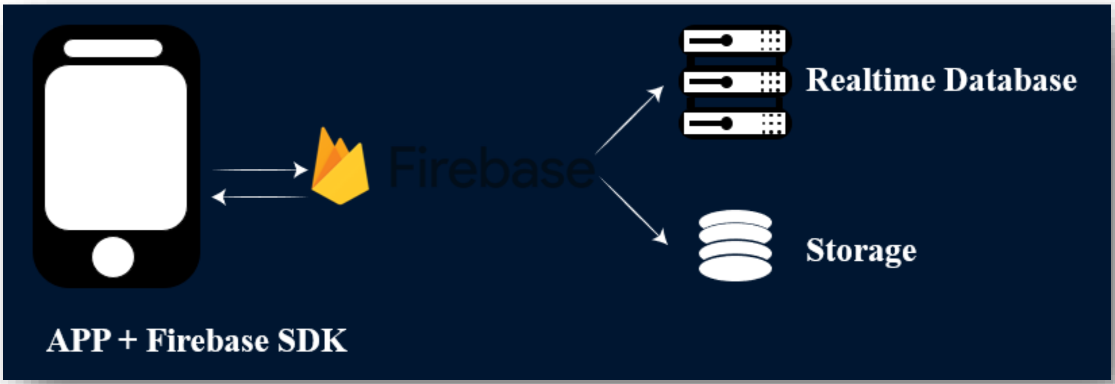

Fig.34. Firebase APP Development.

- **Firebase.**

Firebase is a powerful platform for your mobile and web application. Firebase can power your app’s backend, including data storage, user authentication, static hosting, and more.

Firebase Services?

Fig.35. Firebase Services.

- **Cloud Firestore:**
  is a flexible, scalable database for mobile, web, and server development from Firebase and Google Cloud Platform. It is a NoSQL document database that lets you easily store, sync.

- **ML Kit:**
  is a mobile SDK that brings Google's machine learning expertise to Android and iOS apps in a powerful yet easy-to-use package. Whether you're new or experienced in machine learning, you can implement the functionality you need in just a few lines of code. There's no need to have deep knowledge of neural networks or model optimization to get started. On the other hand, if you are an experienced ML developer, ML Kit provides convenient APIs that help you use your custom TensorFlow Lite models in your mobile apps.

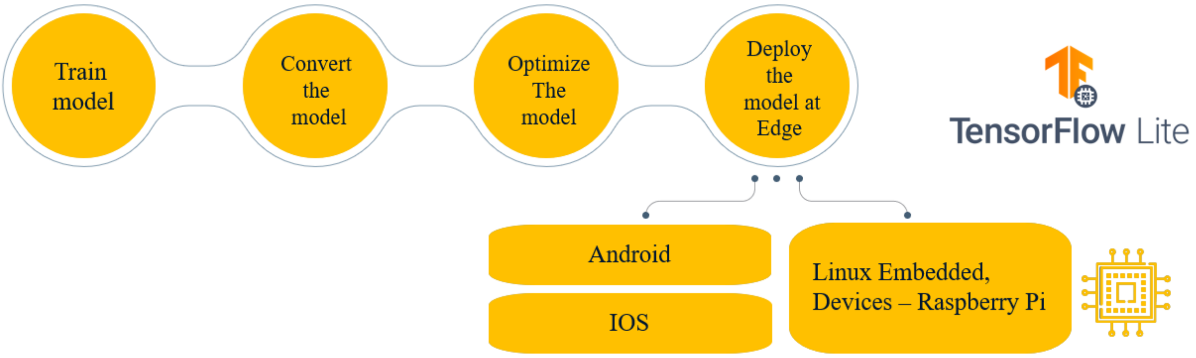

Fig.36. Deploy Deep Learning Model (Flutter Stage).

- **Authentication:** provides backend services, easy-to-use SDKs, and ready-made UI libraries to authenticate users to your app. It supports authentication using passwords, phone numbers, popular federated identity providers like Google, Facebook and Twitter, and more.

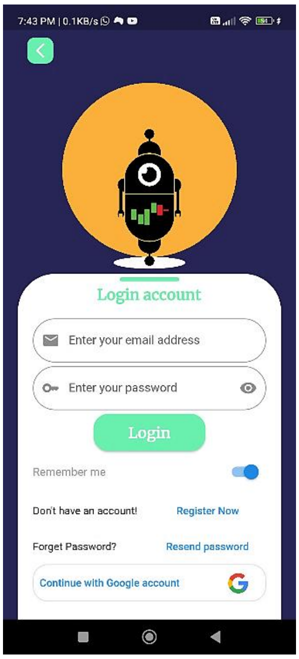

Fig.37. Application Screen (Authentication Required)

- **Real-time Database:** is a cloud-hosted NoSQL database that lets you store and sync between your users in real-time. The Real-time Database is really just one big JSON object that the developers can manage in real-time.

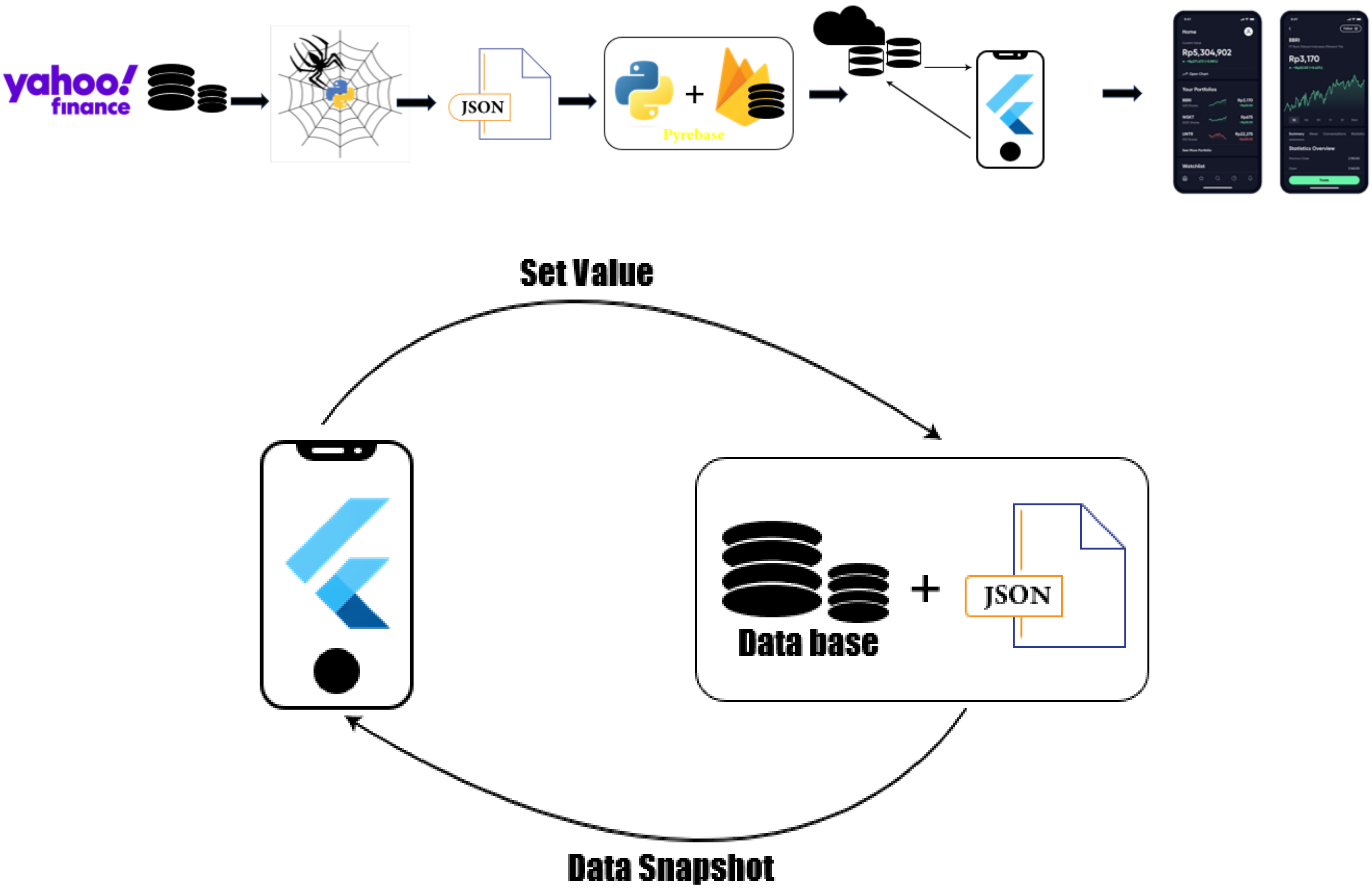

Fig.38. Real-Time Database

Chapter 5

Results and Discussion

Results and Discussion

We have built a system of two parts, a part of artificial intelligence that predicts future stock prices, and a part of the mobile system that monitoring stocks price movements with the possibility of analyzing them using the available analysis methods, with the possibility of buying and selling shares and presenting them to the user. This part was linked to our developed artificial intelligence model, and now we will display the application's screens.

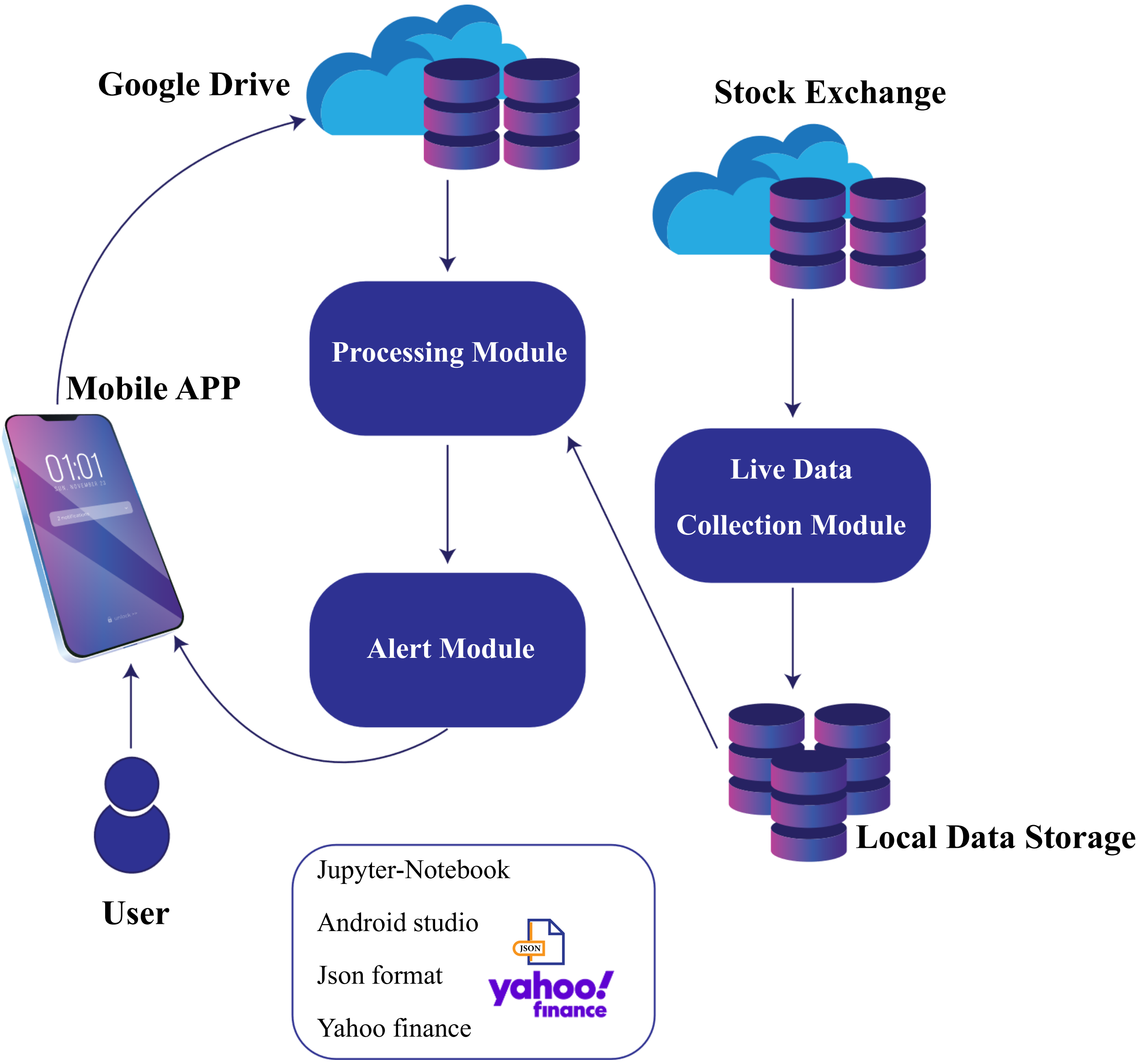

Fig.39. System Structure

Phase-I

- **Data Preprocessing and Deep Learning Model Results.**

statistical analysis of our dataset:

Descriptive statistics include those that summarize the central tendency, dispersion and shape of a dataset's distribution.

For numeric data, the result's index will include count, mean, std, min, max as well as lower, 50 and upper percentiles. By default the lower percentile is 25 and the upper percentile is 75. The 50 percentile is the same as the median.

	open	high	low	close	volume
count	216883.0	216883.0	216883.0	216883.0	216883.0
mean	119.0	119.0	119.0	119.0	11230.0
std	11.0	11.0	11.0	11.0	25509.0
min	94.0	94.0	94.0	94.0	0.0
25%	109.0	109.0	109.0	109.0	3896.0
50%	117.0	117.0	117.0	117.0	6786.0
75%	130.0	130.0	130.0	130.0	11751.0
max	141.0	141.0	141.0	141.0	2110106.0

Table.1. Statistical analysis of dataset

- **Correlation Coefficient between features of dataset:**

	open	high	low	close	volume
open	1.000000	0.999976	0.999972	0.999972	-0.030462
high	0.999976	1.000000	0.999947	0.999978	-0.029033
low	0.999972	0.999947	1.000000	0.999971	-0.031974
close	0.999972	0.999978	0.999971	1.000000	-0.030500
Volume	-0.030462	-0.029033	-0.031974	-0.030500	1.000000

Table.2. Correlation Coeff between the features of dataset.

- **Data Visualization:** is the representation of data through use of common graphics, such as charts, plots, infographics, and even animations.

- **moving average:** is a technical indicator that market analysts and investors may use to determine the direction of a trend. It sums up the data points of a financial security over a specific time period and divides the total by the number of data points to arrive at an average. It is called a "moving" average because it is continually recalculated based on the latest price data.

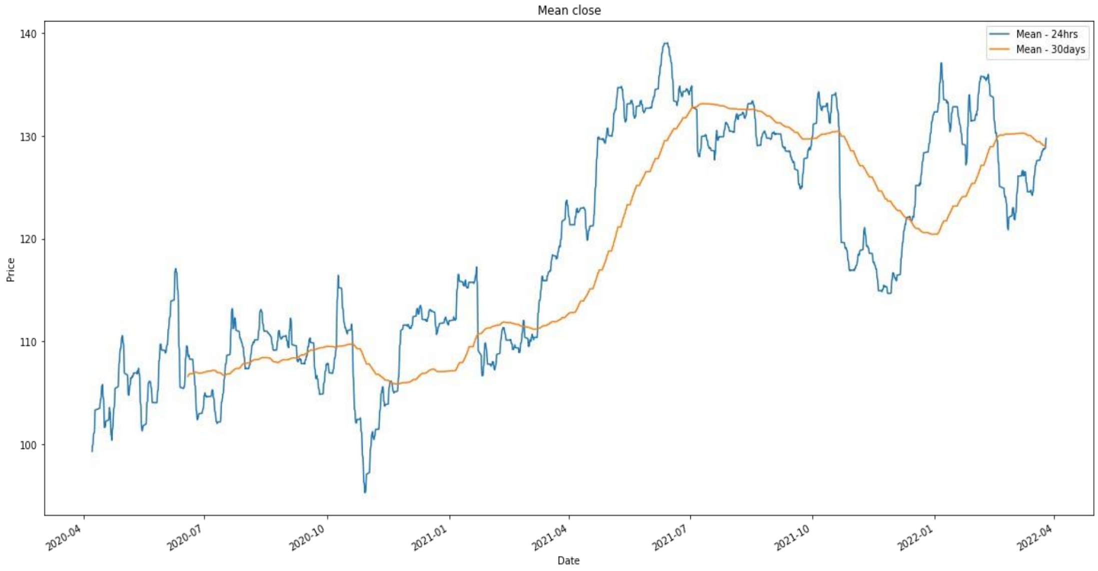

Fig.40. moving average 24 hours & 30 days.

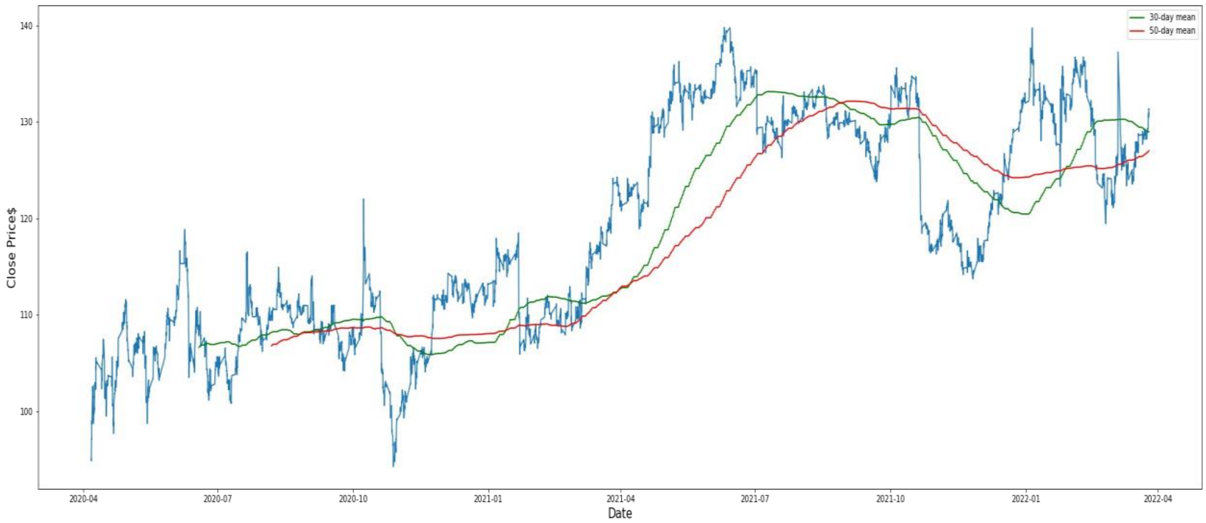

Fig.41. moving average 30 days & 50 days.

- Statistical Analysis.

box plot: displays the five-number summary of a set of data. The five-number summary is the minimum, first quartile, median, third quartile, and maximum. In a box plot, we draw a box from the first quartile to the third quartile.

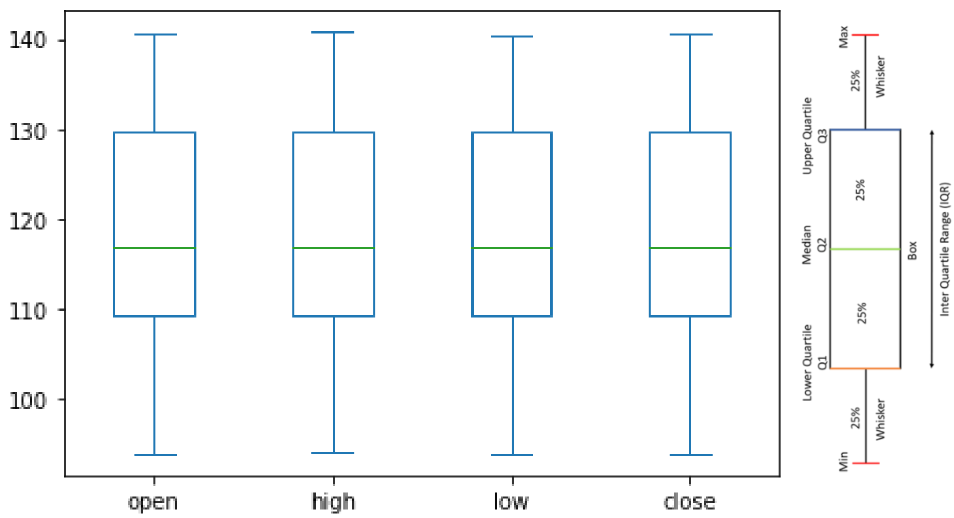

Fig.42. Box Plot (**Statistical Analysis of Data**).

- ### Outliers:

  Outliers are *data* points that are far from other *data* points and they can distort statistical results.

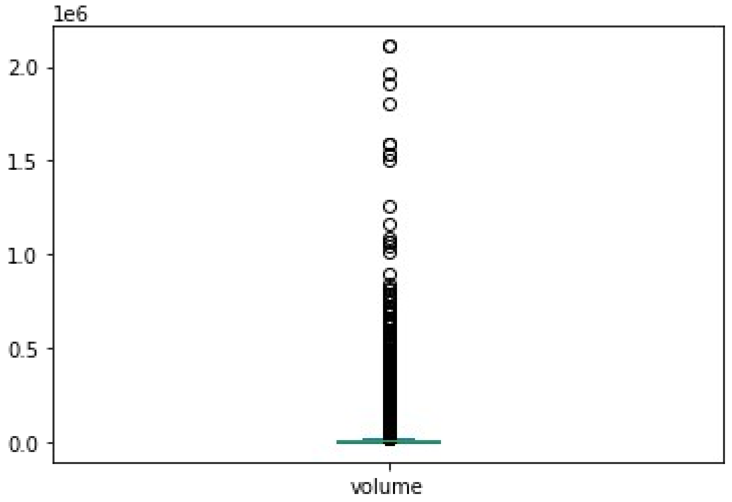

Fig.43. The first step of the outliers search **(box plot).**

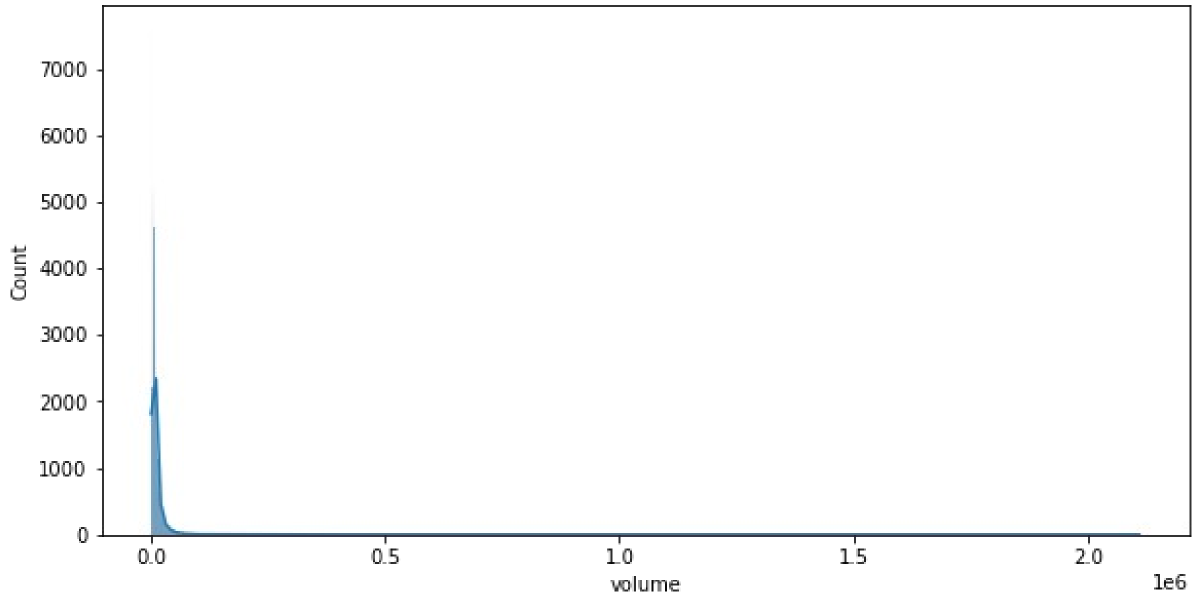

Fig.44. The first step of the outliers search **(histogram).**

Log Transformation:

"Using the log transformation to make data conform to normality."

The normal distribution is widely used in basic and clinical research studies to model continuous outcomes. Unfortunately, the symmetric bell-shaped distribution often does not adequately describe the observed data from research projects. Quite often data arising in real studies are so skewed that standard statistical analyses of these data yield invalid results. Many methods have been developed to test the normality assumption of observed data. When the distribution of the continuous data is non-normal, transformations of data are applied to make the data as "normal" as possible and, thus, increase the validity of the associated statistical analyses. The log transformation is, arguably, the most popular among the different types of transformations used to transform skewed data to approximately conform to normality.

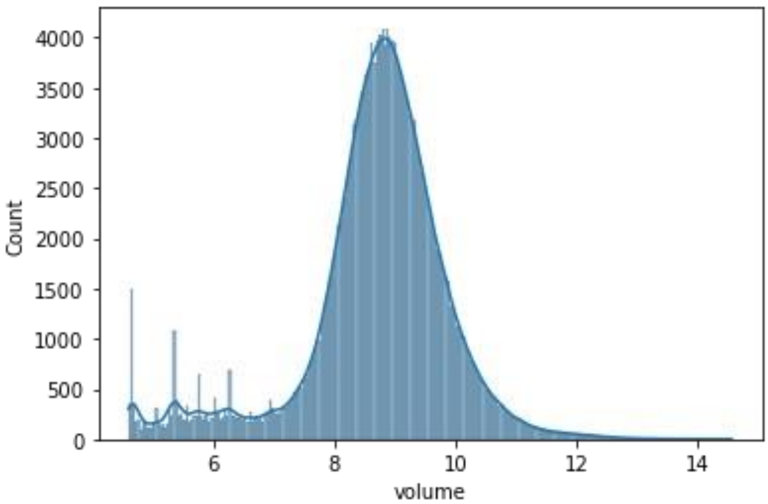

Fig.45. The third step of the outliers search **(Log transformation).**

Now we can clearly see the outliers of data on the box plot figure and take appropriate action.

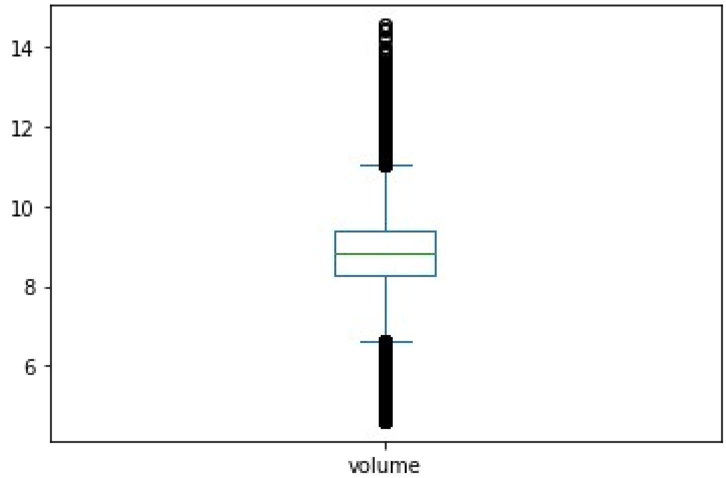

Fig.46. The Last step of the outliers search **(box plot).**

- **Correlation coefficient representation.**

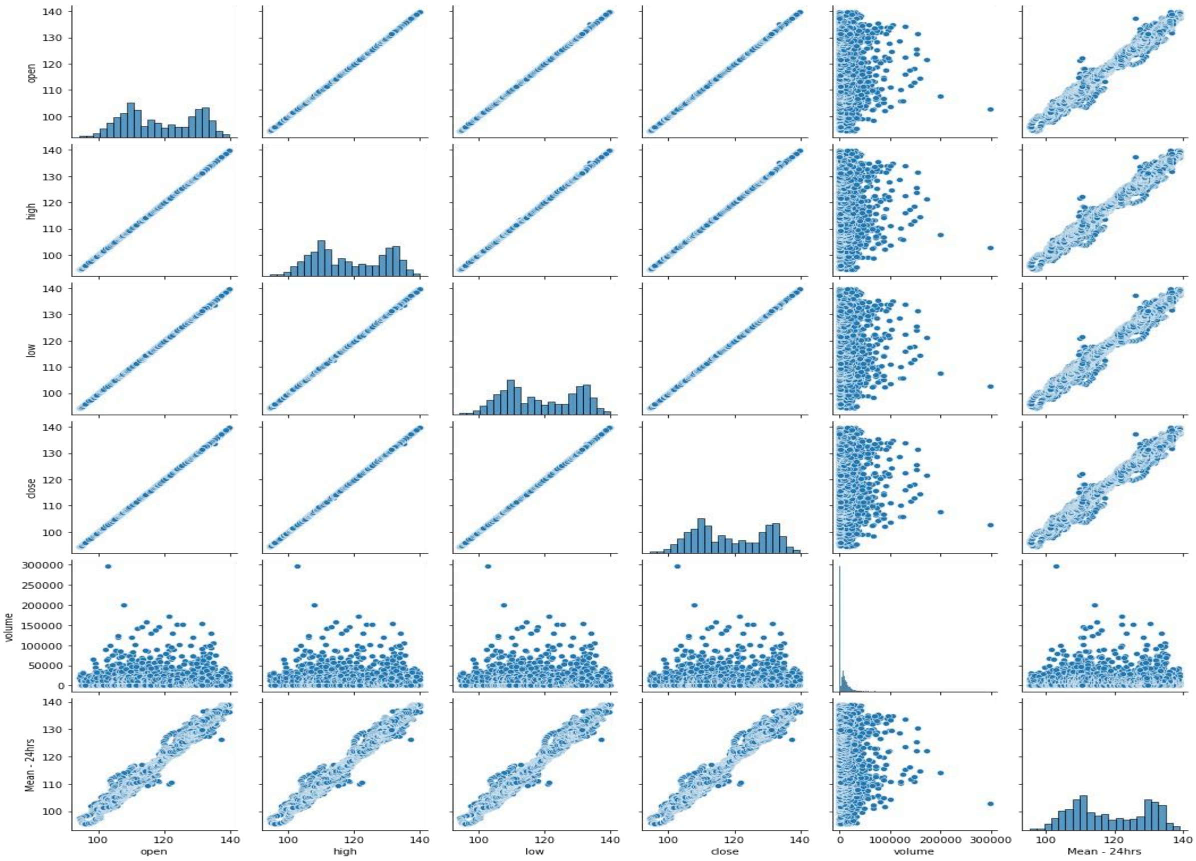

Fig.47. Correlation Matrix.

- **Deep Learning Model Results:**
- **Model Summary:**

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 bidirectional (Bidirectiona  (None, 1, 200)           61800
 l)

 dropout (Dropout)           (None, 1, 200)            0

 lstm (LSTM)                 (None, 1, 100)            120400

 dropout_1 (Dropout)         (None, 1, 100)            0

 lstm_1 (LSTM)               (None, 1, 50)             30200

 dropout_2 (Dropout)         (None, 1, 50)             0

 lstm_2 (LSTM)               (None, 50)                20200

 dropout_3 (Dropout)         (None, 50)                0

 dense (Dense)               (None, 1)                 51

=================================================================
Total params: 232,651
Trainable params: 232,651
Non-trainable params: 0
_________________________________________________________________
None
```

Fig.48. Model Summary.

- **Layers and number of params in each layer:**

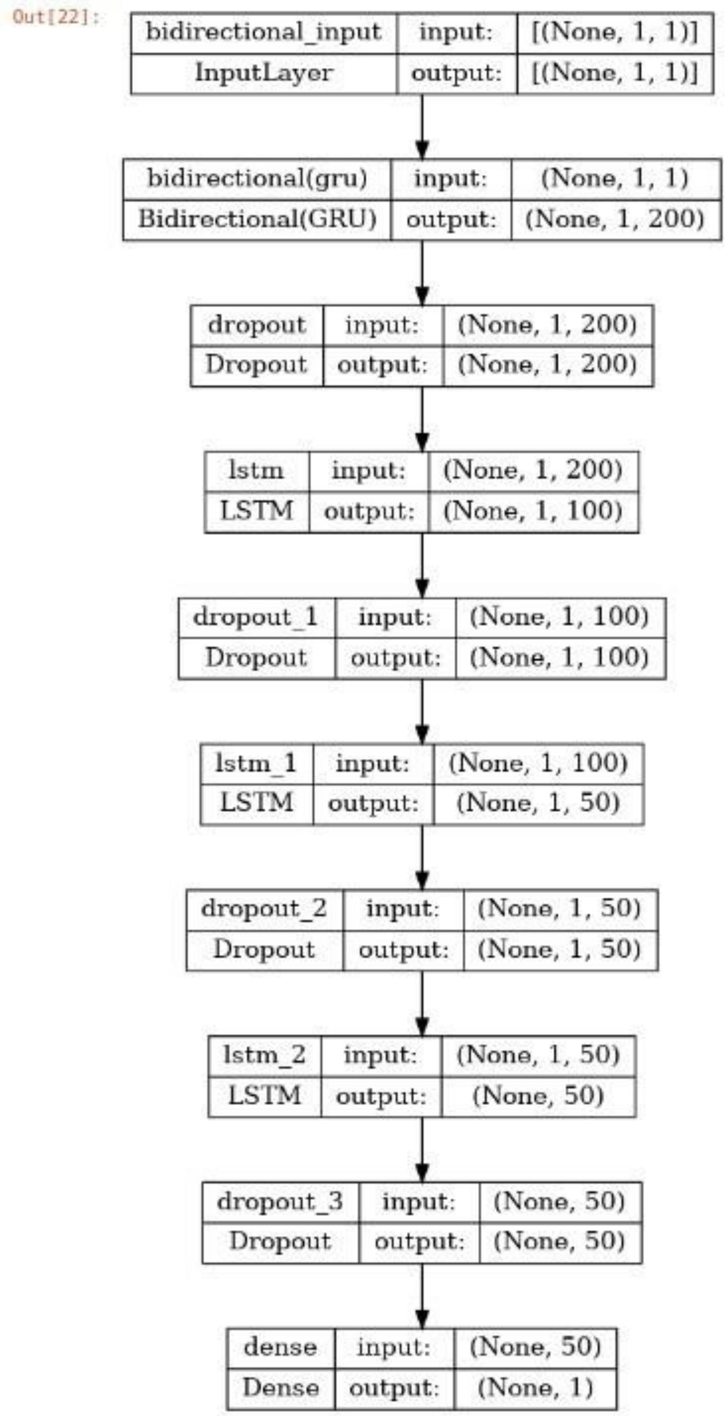

Fig.49. Model Layers and number of params in each layer.

- **Bi GRU-LSTM Model Results:**

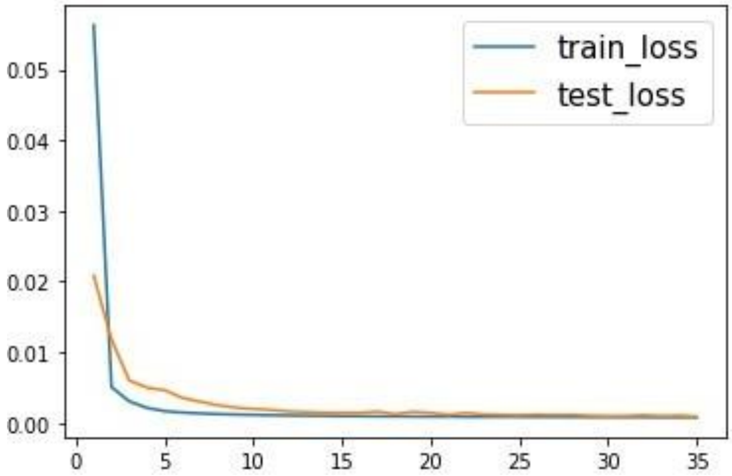

Fig.50. train loss vs test loss (Bi GRU-LSTM)

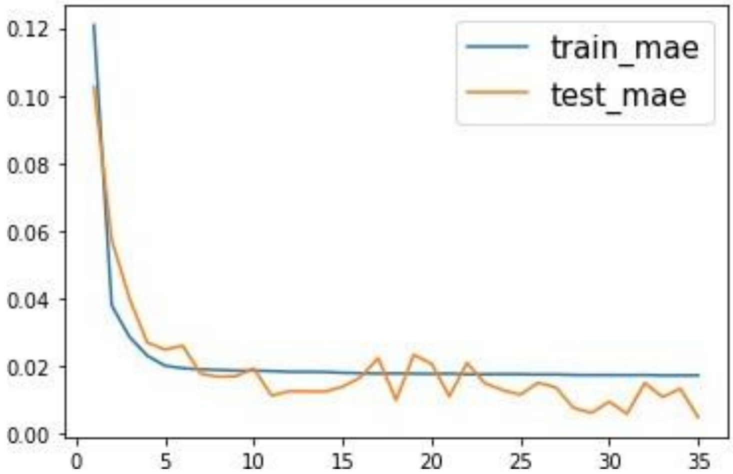

Fig.51. train MAE vs test MAE (Bi GRU-LSTM)

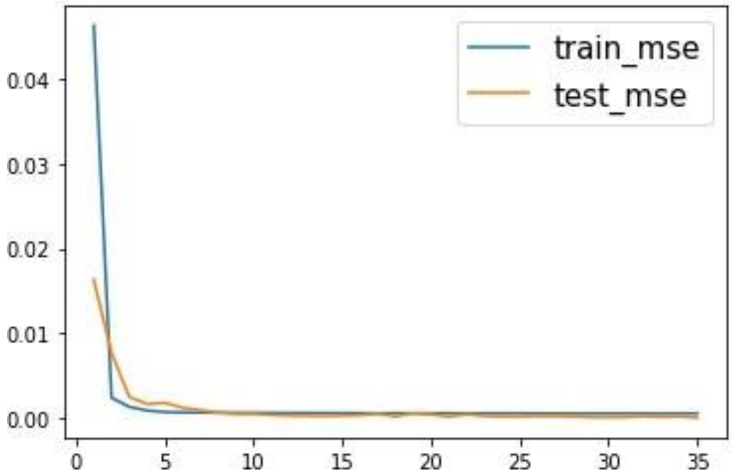

Fig.52. train MSE vs test MSE (Bi GRU-LSTM)

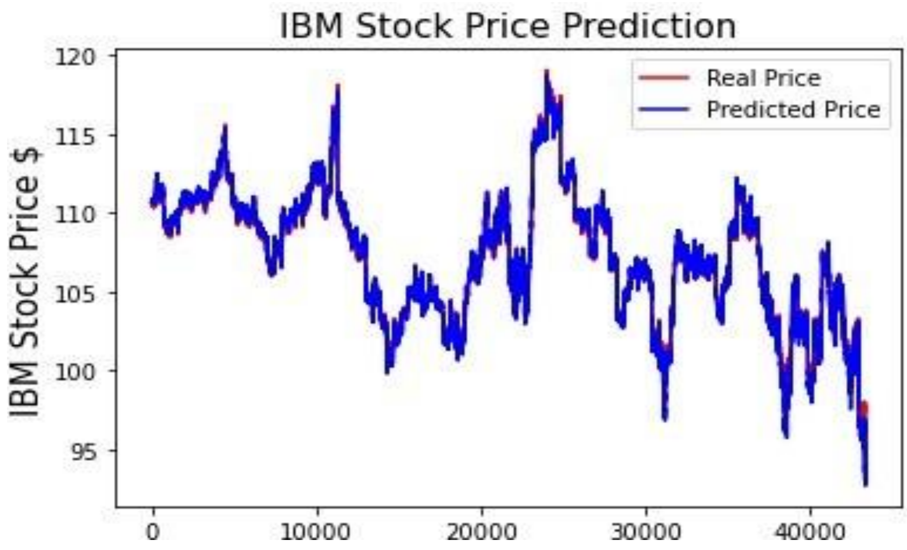

Fig.53. IBM Stock Actual Price vs Predicted Price

(Bi GRU-LSTM)

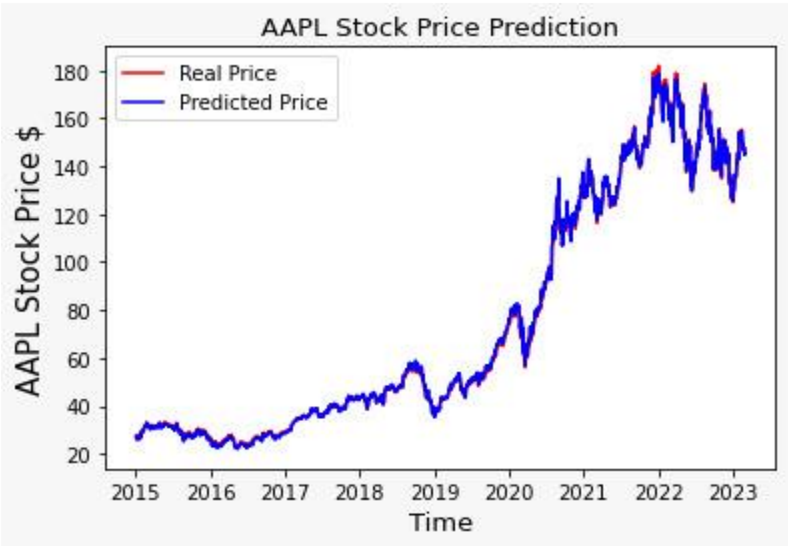

Fig.54. AAPL Stock Actual Price vs Predicted Price

(Bi GRU-LSTM)

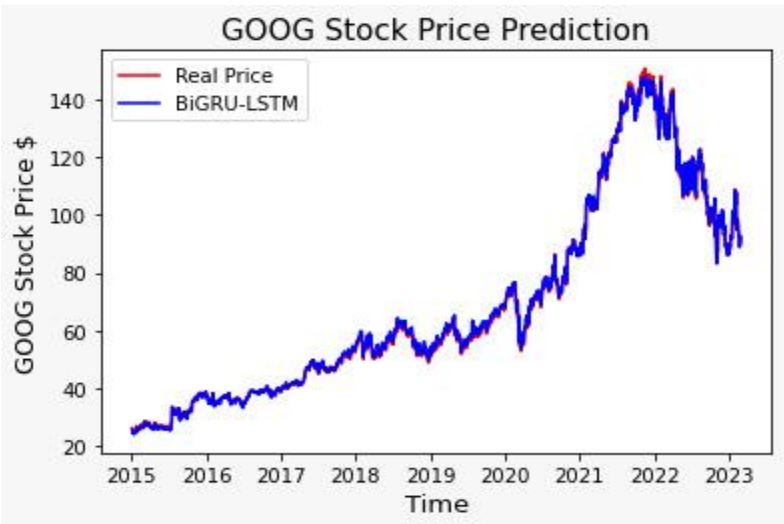

Fig.55. GOOG Stock Actual Price vs Predicted Price

(Bi GRU-LSTM)

(Bi GRU-LSTM) Model Evaluation:

testing results:

```
R2: 0.9942, MAE: 0.2073, MSE: 0.0918
```

training results:

```
R2: 0.9990, MAE: 0.0047, MSE: 4.5065e-05
```

- **GRU-LSTM Model Results:**

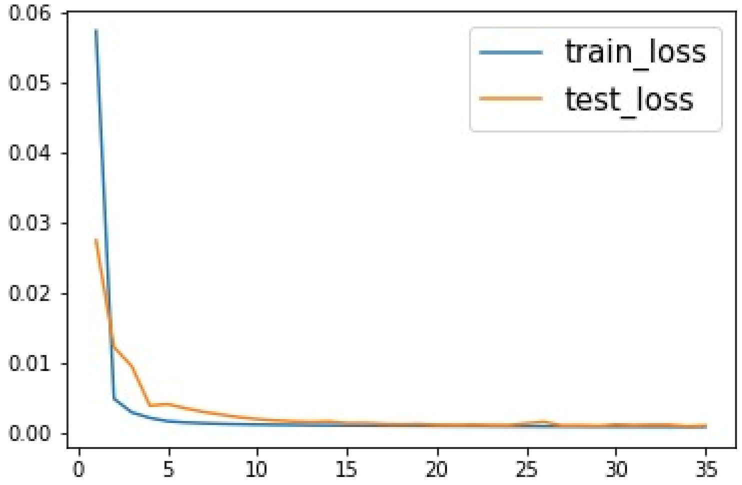

Fig.56. train loss vs test loss (GRU-LSTM)

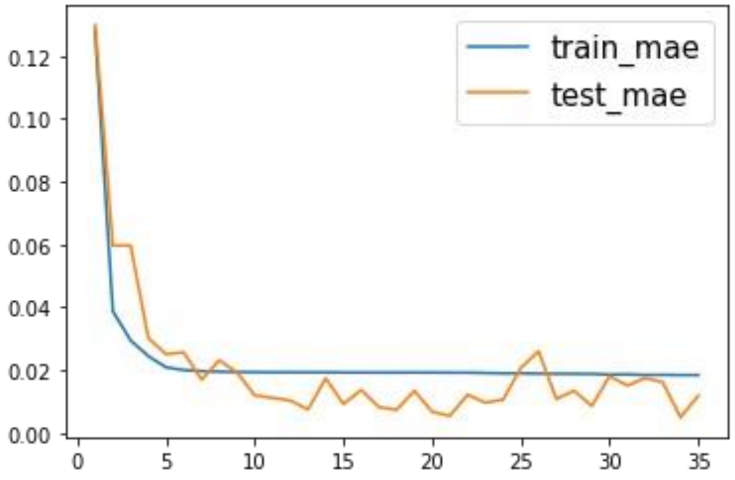

Fig.57. train MAE vs test MAE (GRU-LSTM)

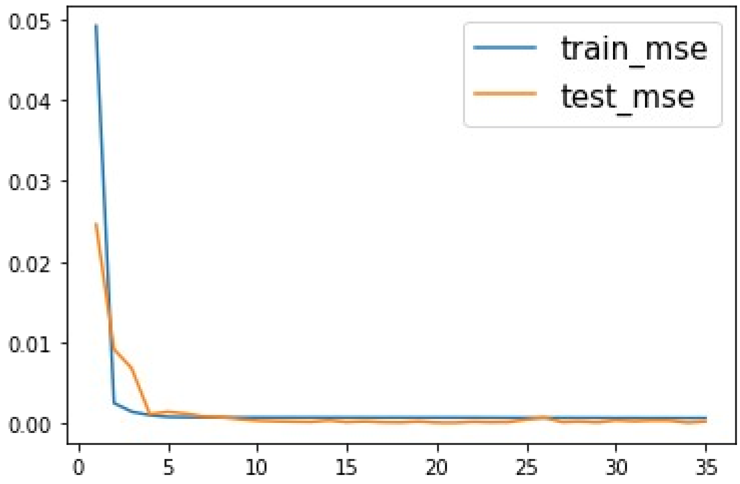

Fig.58. train MSE vs test MSE (GRU-LSTM)

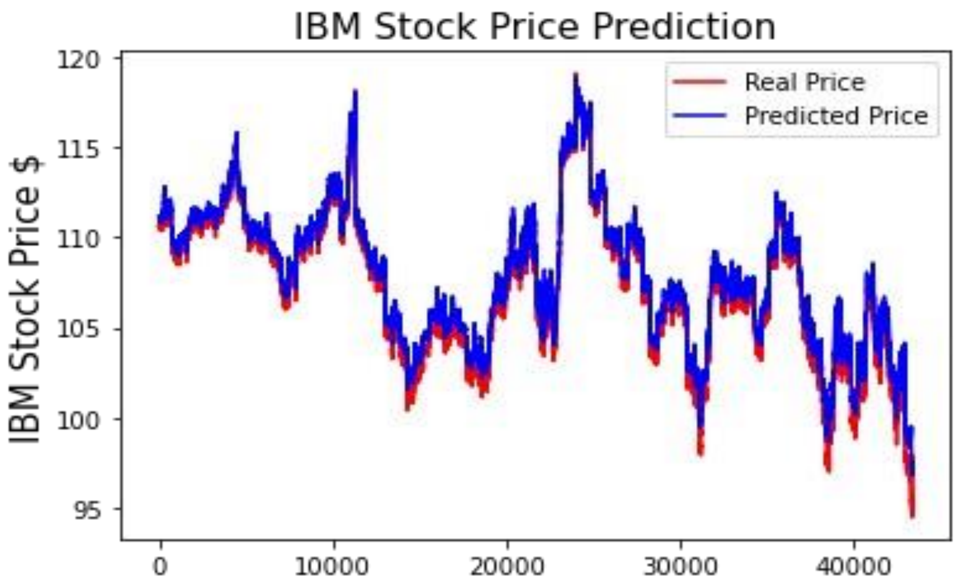

Fig.59. IBM Stock Actual Price vs Predicted Price (GRU-LSTM)

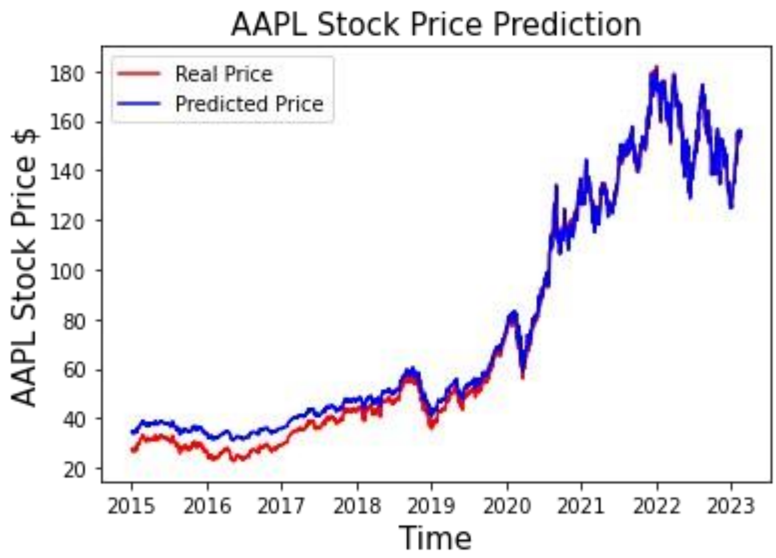

Fig.60. AAPL Stock Actual Price vs Predicted Price

(GRU-LSTM)

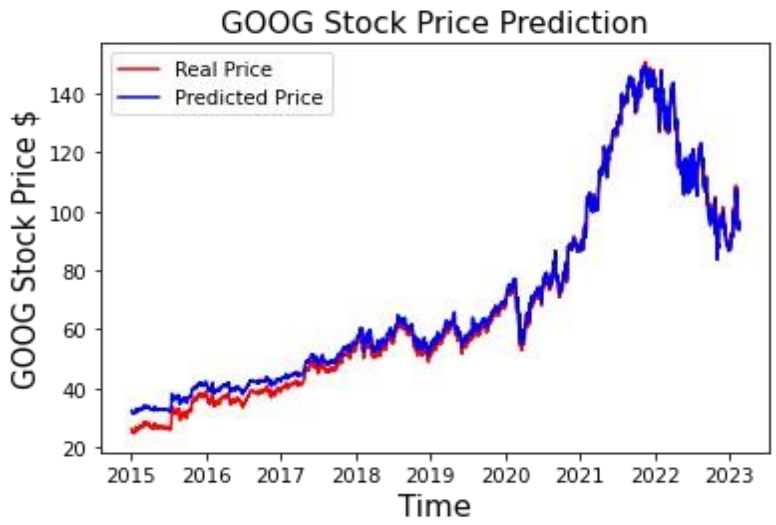

Fig.61. GOOG Stock Actual Price vs Predicted Price

(GRU-LSTM)

(GRU-LSTM) Model Evaluation:

testing results:

`R2: 0.9707 , MAE: 0.6277 , MSE: 0.4694`

training results:

`R2: 0.9981 , MAE: 0.0075 , MSE: 8.9253e-05`

- **LSTM Model Results:**

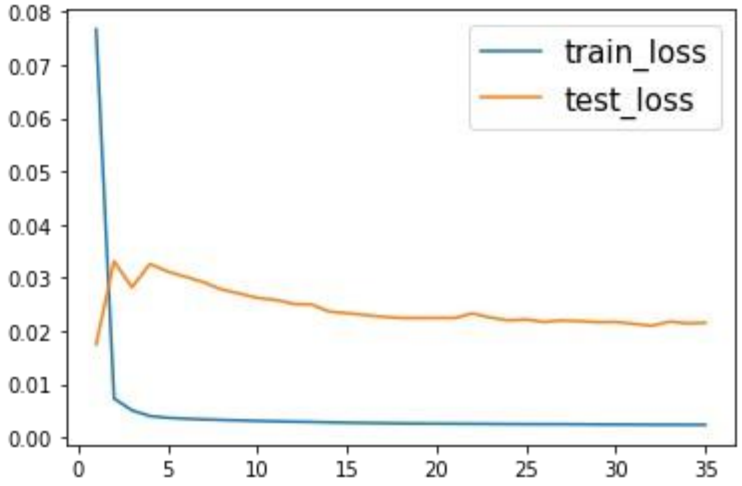

Fig.62. train loss vs test loss (LSTM)

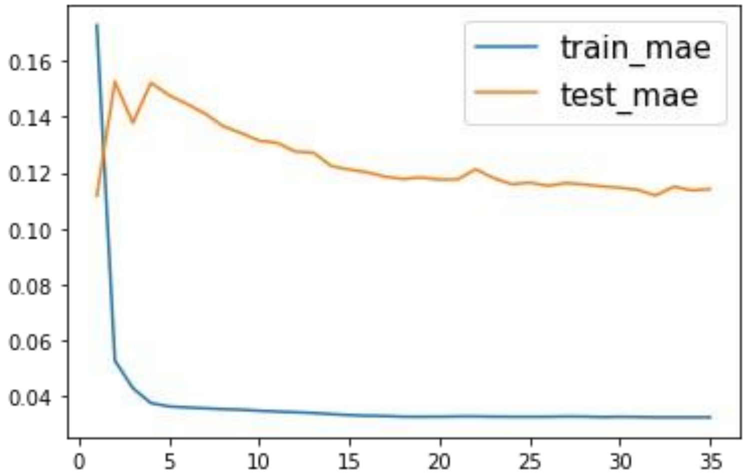

Fig.63. train MAE vs test MAE (LSTM)

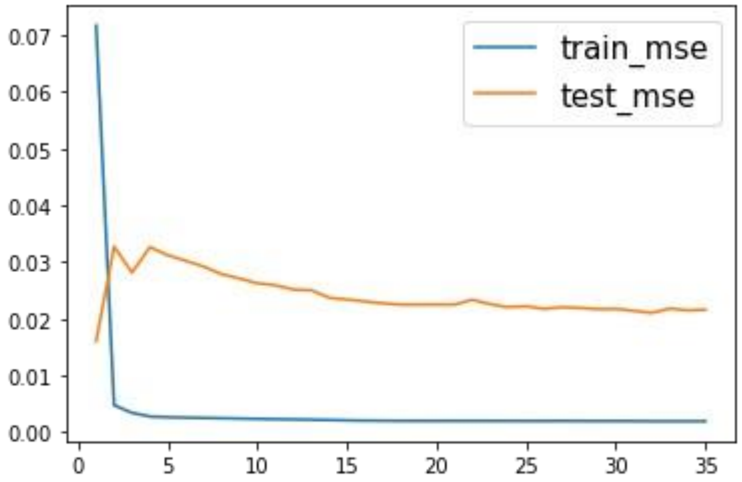

Fig.64. train MSE vs test MSE (LSTM)

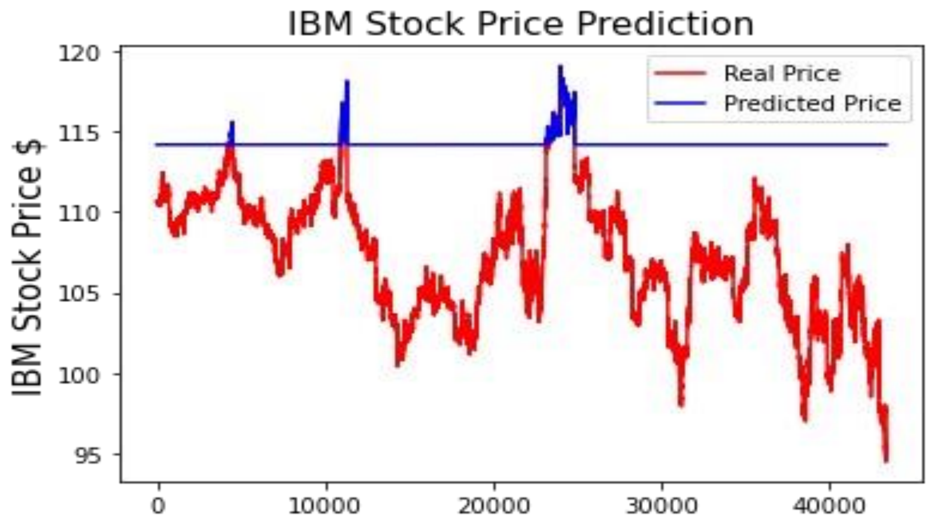

Fig.65. IBM Stock Actual Price vs Predicted Price (LSTM)

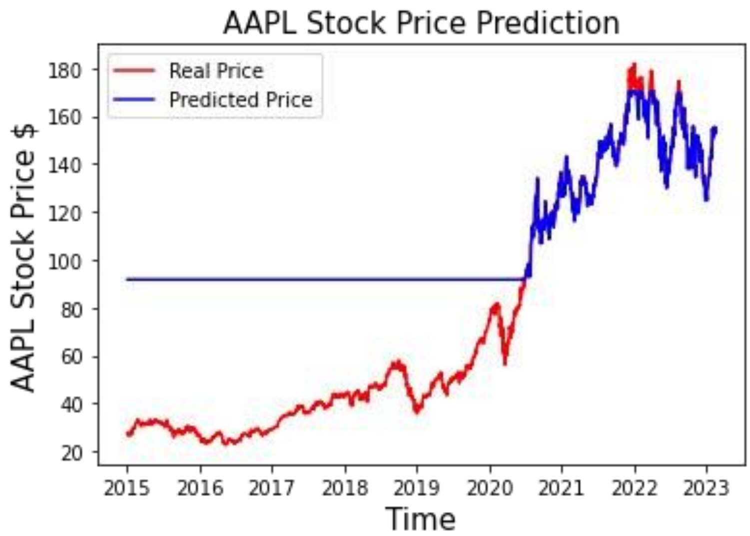

Fig.66. AAPL Stock Actual Price vs Predicted Price (LSTM)

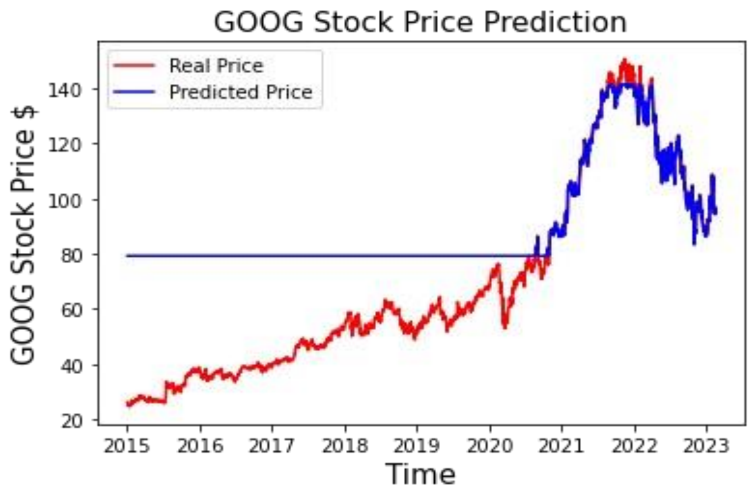

Fig.67. GOOG Stock Actual Price vs Predicted Price (LSTM)

(LSTM) Model Evaluation:

testing results:

```
R2: -2.9528, MAE: 6.9952, MSE: 63.4849

training results:

R2: 0.8624, MAE: 0.0394, MSE: 0.0068
```

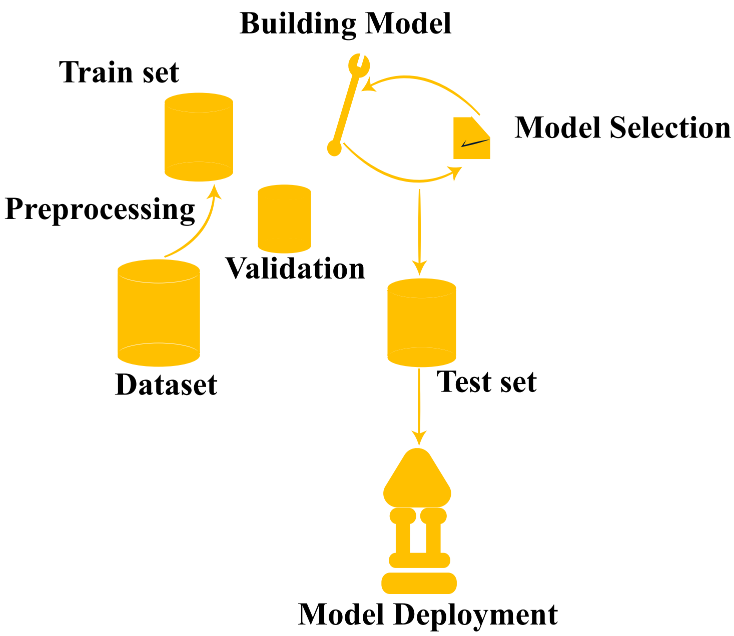

Fig.68. All Model Processes.

Phase II

- **Mobile Application Results:**

- **Application Intro Screens.**

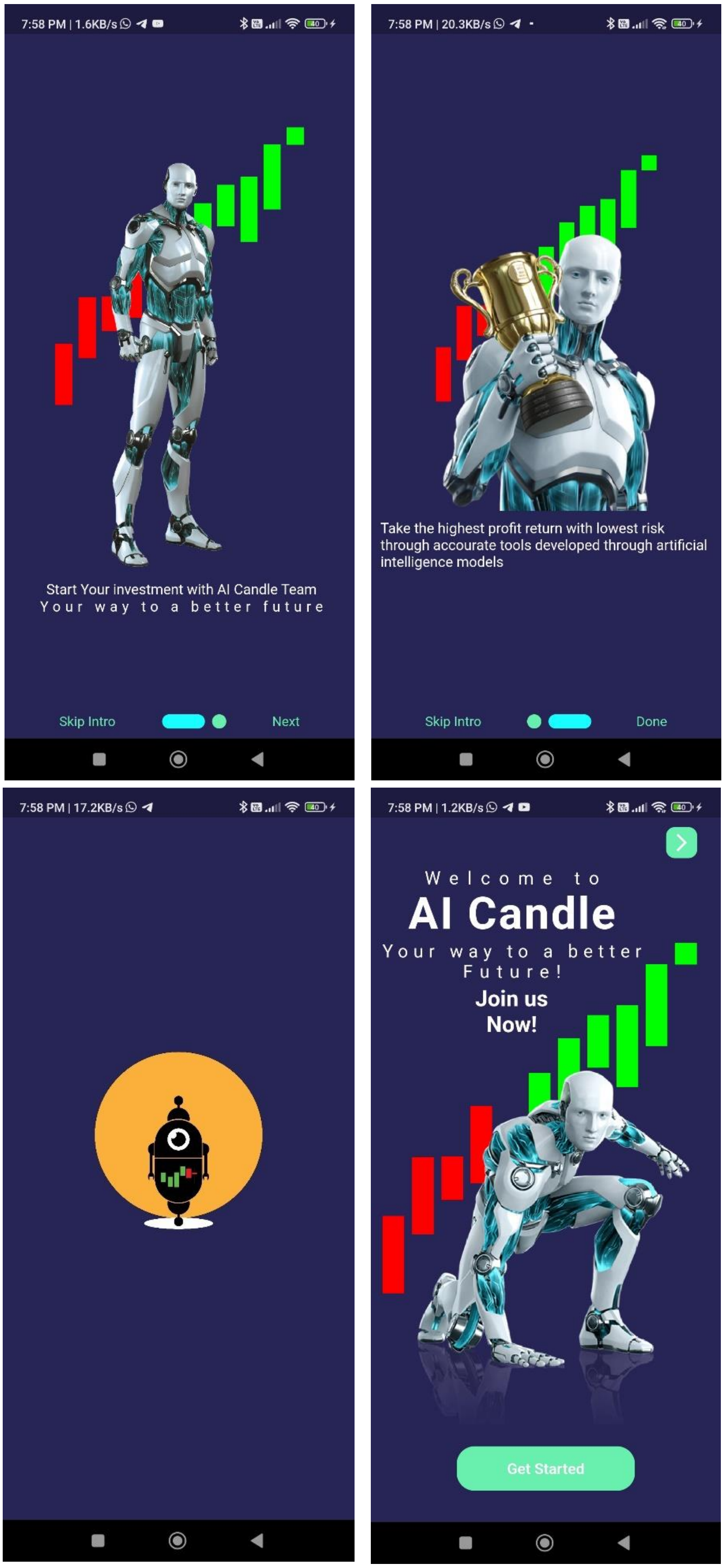

Fig.69. APP Intro Screens.

- **Login Layout.**

The fourth page that comes to users after installing the applications with these facilities:

Register Now: first time to use app. And to create new account.

Login: if user already have an account whether if as admin or investor.

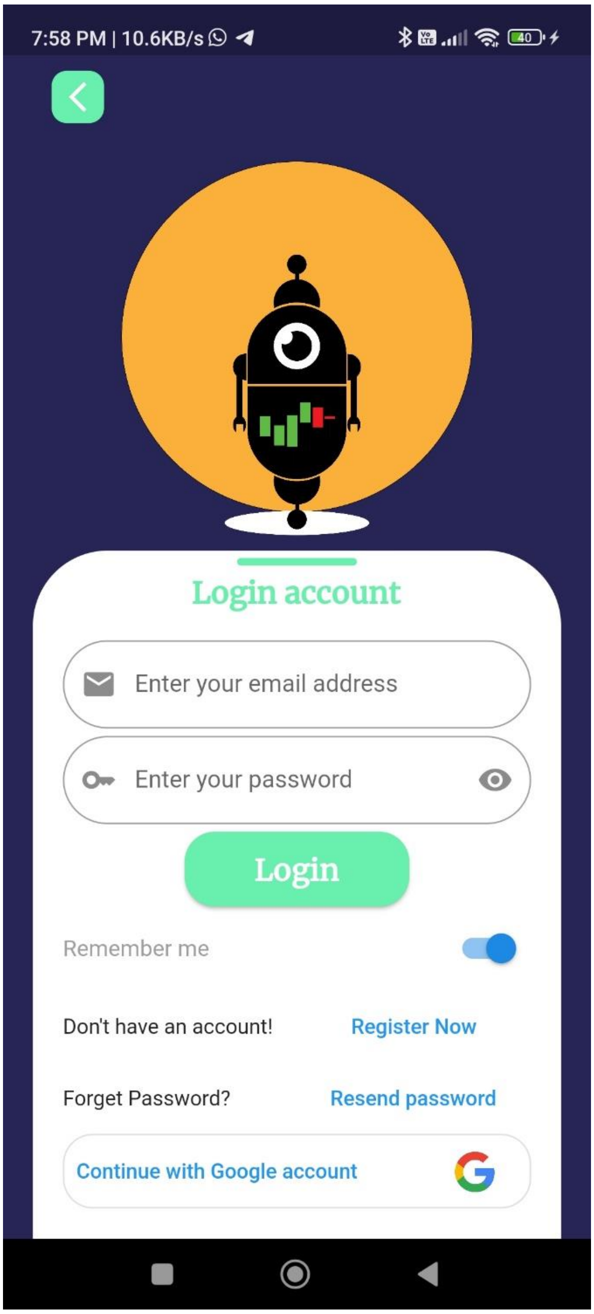

Fig.70. Login Screen.

- **Register Now Layout.**

  It requires user data to be stored at database like: first name, last name, mobile number, e-mail and password.

  If user type is admin, then a system verification code must be entered.

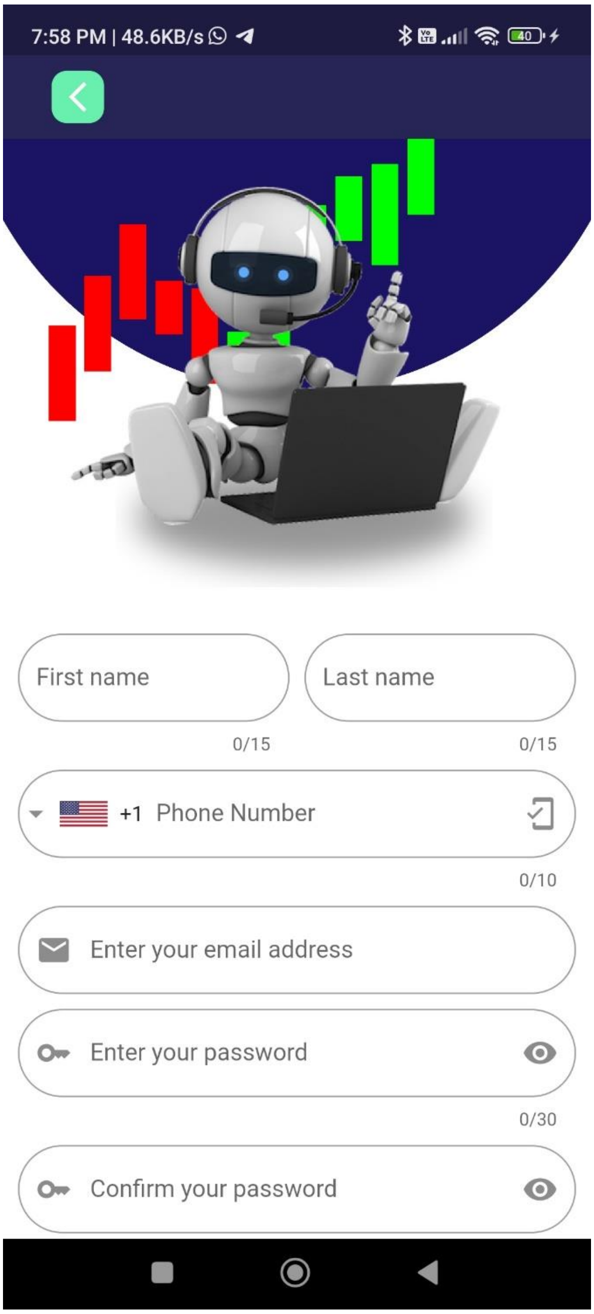

Fig.71. Register Now Screen.

- **Home Layout.**

Here, there are the stocks that available for the users to trade on and the profit and loss rate for each stock.

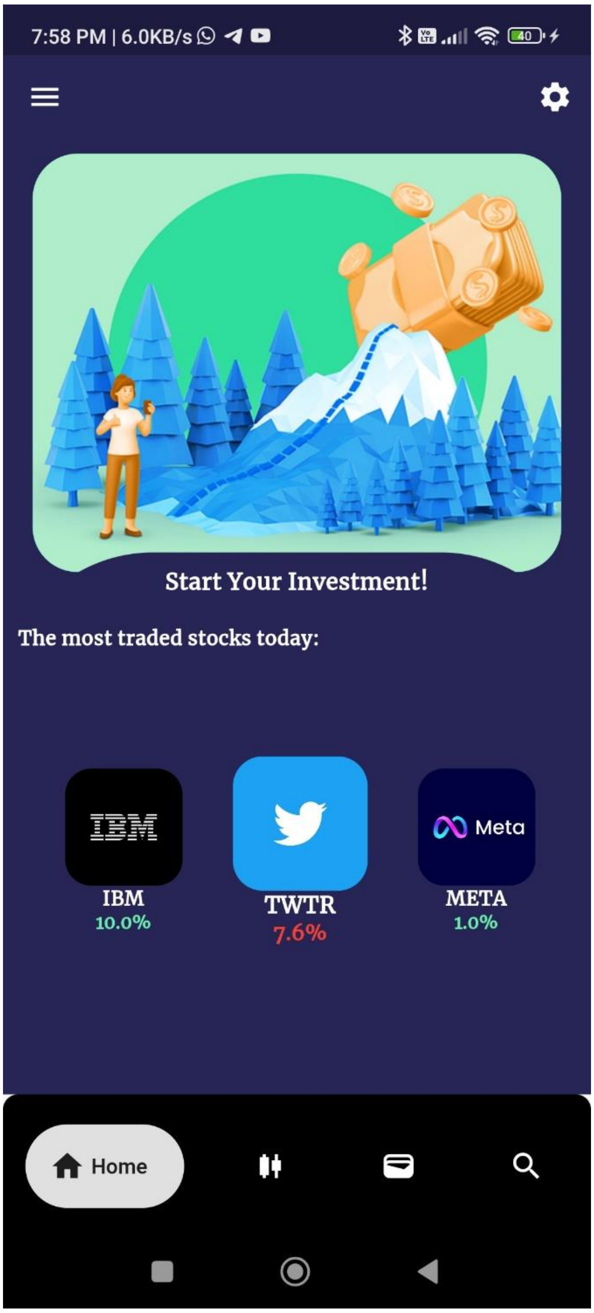

Fig.72. Home Screen.

- **Stock state Layout.**

  Here we display a line chart of the stock price over the day, week and month frames.

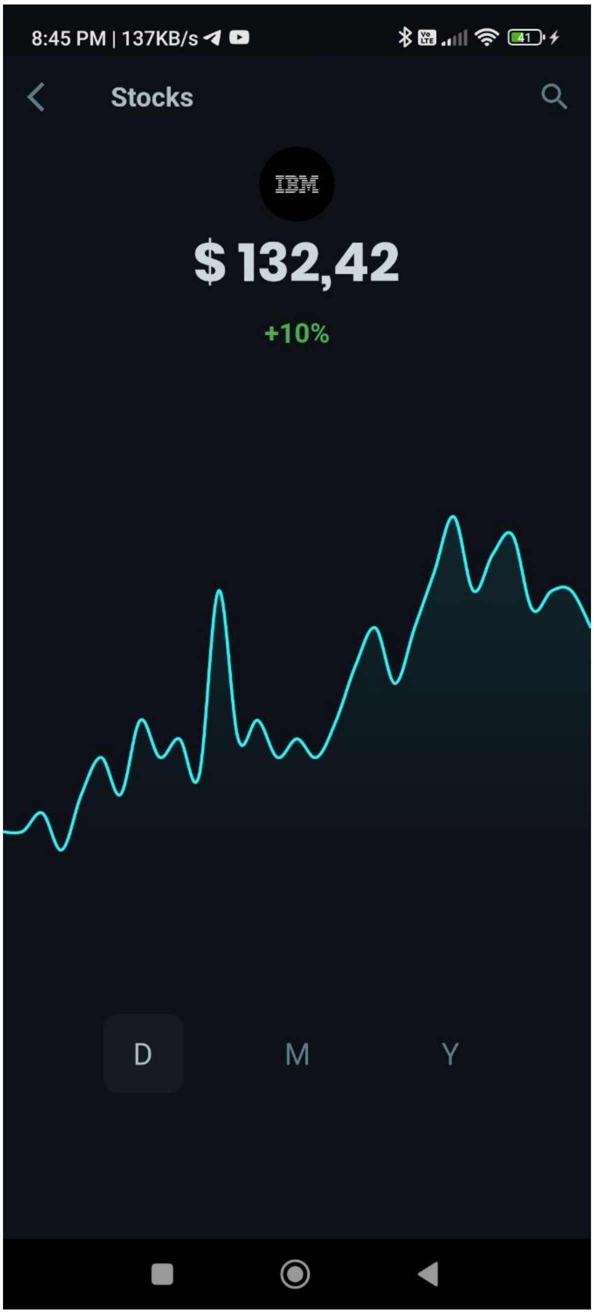

Fig.73. Stock state Screen (Line graph).

- **Candle stick chart layout.**

Here we display a candle stick chart and some technical analysis tools that help the user to make appropriate decisions in the trading process on the minute, hour, day, week and month frames.

Fig.74. Candle stick chart Screen.

- **Wallet Screen.**

  Here we give the user many permissions to link different bank accounts, withdraw and deposit money, to be able to buy and sell stocks.

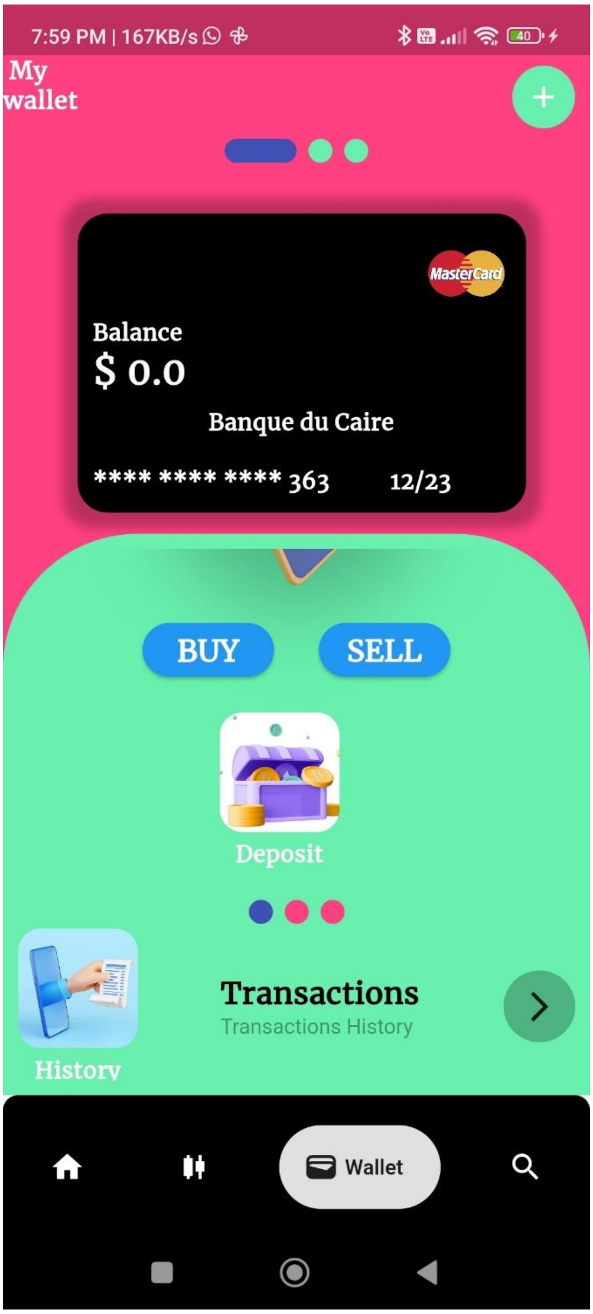

Fig.75. Wallet Screen.

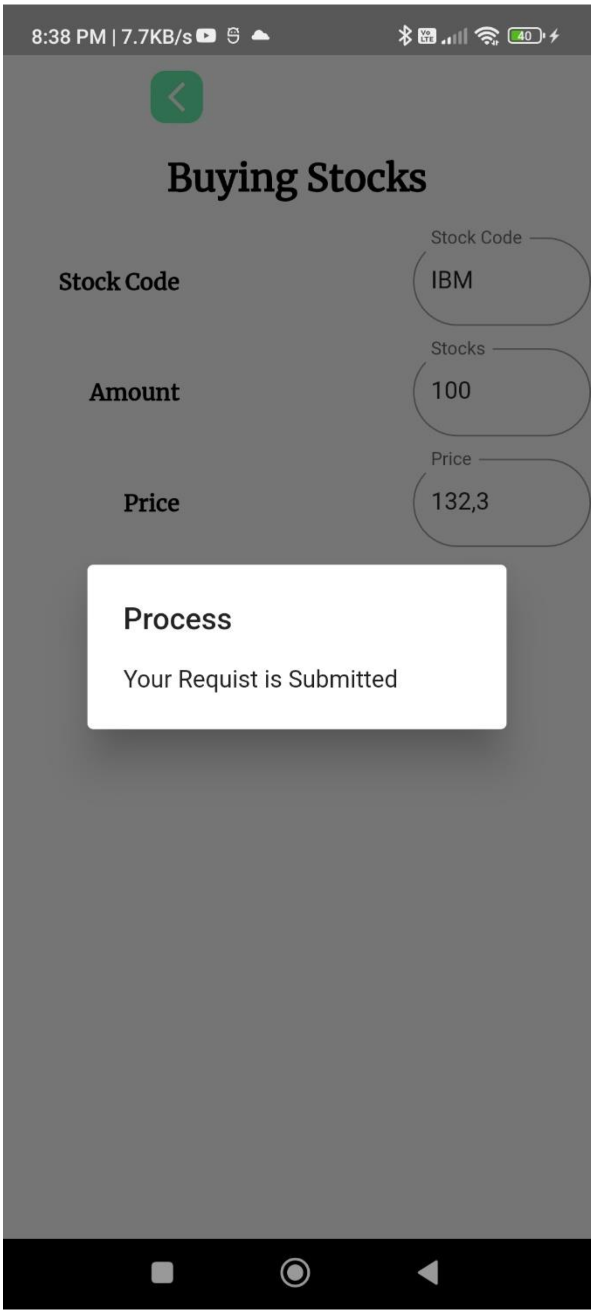

Fig.76. Buying stocks screen.

- **Analysis Layout.**

  This section is concerned with the future prediction of prices, support prices, and various resistances for the stock.

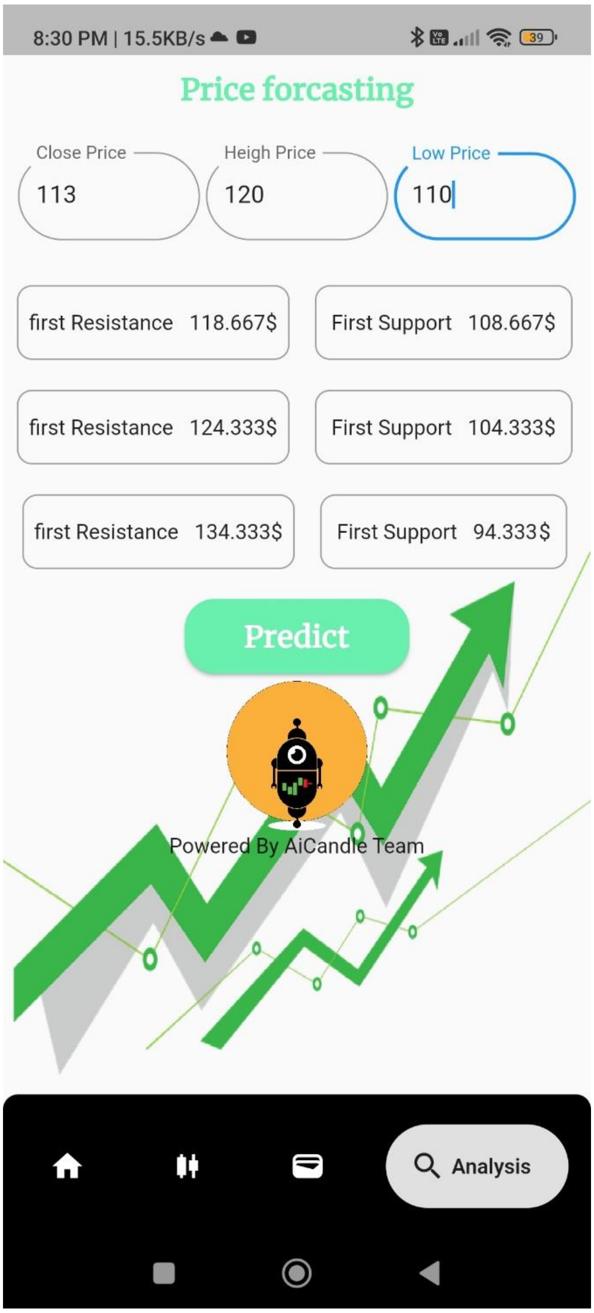

Fig.77. Analysis Phase.

- **Other application features.**
  Other features related to the application, such as contacting us, searching for favorite stocks, and settings…etc.

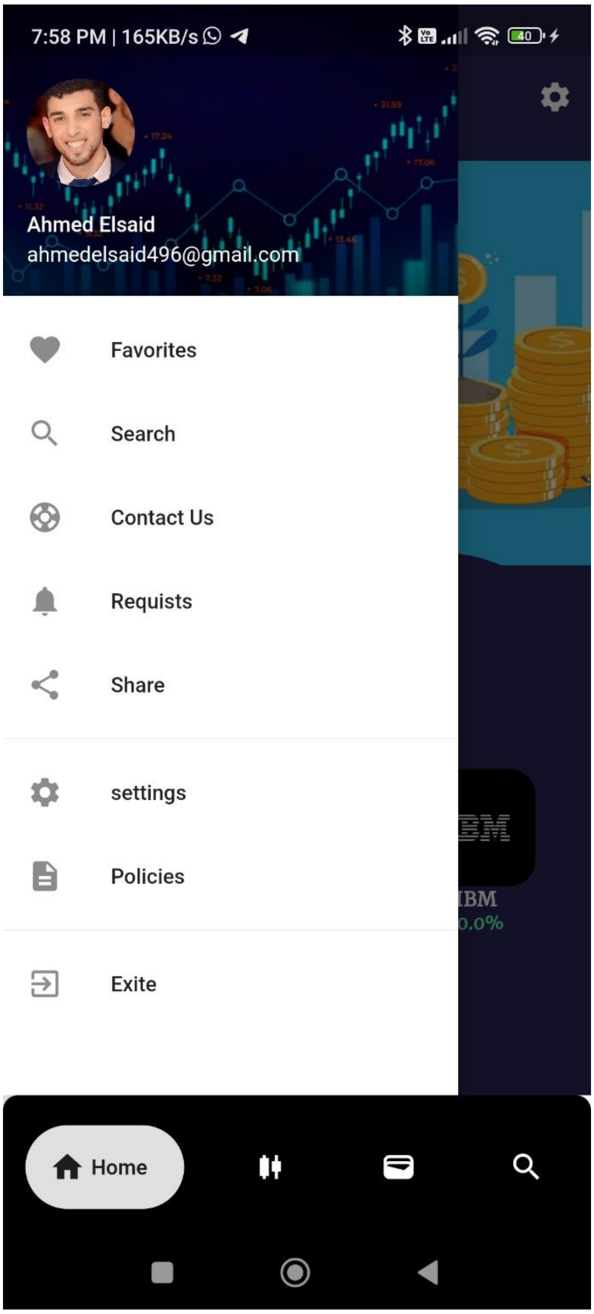

Fig.78. APP features.

Chapter 6

Conclusion & Future Work

Conclusion and Future Work

6.1. Conclusion.

To develop the Deep Learning/ (**IEEE 2941-2021**) prediction model, the implementation process should be gone through relevant data collection, data preprocessing to remove noise and missing values. Analyzing the best Algorithm followed by model evaluation. This Project uses the Recurrent Neural Network (RNN) with BiGRU-LSTM cells to predict the movement of stock market exchange.

The results show that the Deep Learning/ (**IEEE 2941-2021**) model is prone to give more accurate result than the traditional machine learning algorithms.

This model can be proven to be productive for individual traders as well as for corporate investors. They can get the future behavior of market price movement and take the proper action to make a profit.

In future work, the project should be considered different features and aspects of the market to make prediction more accurate. Also, we intend to use reviews of the users on the product to predict the change in the market.

6.2. References.

[1] W., Khan, M., Ghazanfar, M., Azam, *et al.*,"Stock market prediction using machine learning classifiers and social media, news," Journal of Ambient Intelligence and Human Computing, Volume 13, https://doi.org/10.1007/s12652-020-01839-w, 2022, PP. 3433–3456.

[2] 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," *IEEE Access*, Volume 8, doi: 10.1109/ACCESS.2020.3015966, 2020, PP. 150199-150212.

[3] https://sagroups.ieee.org/2941/aim-standards/

[4] M. Nikou, G. Mansourfar, and J. Bagherzadeh," *Stock price prediction using deep learning algorithm and its comparison with machine learning algorithms*," Wiley Online Library, Volume 26, https://doi.org/10.1002/isaf.1459, 2019, PP. 164-174.

[5] X. Pang, Y. Zhou, P. Wang, et al.," Aninnovative neural network approach for stock market prediction," Journal of Supercomputing, Springer, Volume 76, https://doi.org/10.1007/s11227-017-2228-y, 2020, PP. 2098–2118.

[6] W. Jiang," Applications of deep learning in stock market prediction: Recent progress," Expert systems with applications, Elsevier, Volume 184, https://doi.org/10.1016/j.eswa.2021.115537, 2021, PP. 1-22.

[7] R. Singh and Sh. Srivastava," Stock prediction using deep learning," Multimedia tools and Applications, Springer, Volume 76, DOI 10.1007/s11042-016-41Payal Soni 1, Yogya Tewari 1 and Prof. Deepa Krishnan59-7, 2017, PP. 18569–18584.

[8] P. Soni, Y.Tewari, and D. Krishnan," Machine Learning Approaches in Stock Price Prediction: A Systematic Review," Journal of Physics: Conference Series, IOP Publishing, doi:10.1088/1742-6596/2161/1/012065, 2022, PP. 1-10.

[9] C. Stoean, W. Paja, R. Stoean, and A. Sandita, "*Deep architectures for long-term stock price prediction with a heuristic-based strategy for trading simulations*," PLOS One, Volume 14, Issue 10, https://doi.org/10.1371/journal.pone.0223593, 2019.

[10] M. Sedighi, H. Jahangirnia, M. Gharakhani, and S. Farahani," A novel hybrid model for stock price forecasting based on metaheuristics and support vector machine," Data, MDPI, Volume 4, Issue 2, doi:10.3390/data4020075, 2019, PP. 1-28.

[11] M. Nabipour, P. Nayyeri, H. Jabani, et al., "*Deep Learning for Stock Market Prediction*," Entropy, MDPI, Volume 22, Issue 8, 2020, PP. 1-23.

[12] X. Pang, Y. Zhou, P. Wang, et. al.," An innovative neural network approach for stock for stock market prediction," Journal of Supercomputing, Springer, Volume 76, https://doi.org/10.1007/s11227-017-2228-y, 2020, PP. 2098-2118.

[13] X. Li, P. Wu, and W. Wang," *Incorporating stock prices and news sentiments for stock market prediction: A case of Hong Kong*," Image Processing & Management, Elsevier, Volume 57, Issue 5, 2021, PP.

[14] W. Lu, J. Li, J. Wang, and L. Qin," *A CNN-BiLSTM-AM method for stock price prediction*," Neural Computing and Application, Springer, Volume 33, https://doi.org/10.1007/s00521-020-05532-z(0123456789, 2021, PP. 4741–4753.

[15] M. J. Awan, M. Shafry, H. Nobanee, A. Munawar, A. Yasin et al., "*Social media and stock market prediction: a big data approach*," Computers, Materials & Continua, volume 67, Issue 2, 2021, PP. 2569–2583.

[16] R. Jamous, H. ALRahhal, and M. El-Darieby," A new ANN-particle swarm optimization with center of gravity (ANN-psocog) prediction model for the stock market under the effect of COVID-19," Scientific Programming, Hindawi, https://www.hindawi.com/journals/sp/2021/6656150/, 2021, pp. 1– 17,

[17] A. Thakkar, and K. Chaudhari, "*Fusion in stock market prediction: A decade survey on the necessity, recent developments, and potential future directions*," Information Fusion, Elsevier, Volume 65, 2021, PP. 95–107, Elsevier.

[18] Y. Ji, A. W. -C. Liew and L. Yang, "*A Novel Improved Particle Swarm Optimization with Long-Short Term Memory Hybrid Model for Stock Indices Forecast*," in *IEEE Access*, Volume 9, doi: 10.1109/ACCESS.2021.3056713, PP. 23660-23671, 2021.

[19] S. Albahli, T. Nazir, A. Mehmood, et. al.," *AEI-DNET: A Novel DenseNet Model with an Autoencoder for the Stock Market Predictions Using Stock Technical Indicators*," Electronics, MDPI, Volume 11, Issue 4, 2022, https://doi.org/10.3390/electronics11040611, PP.

[20] W. Hussain, J. Merigó, and M. Raza," *Predictive intelligence using ANFIS-induced OWAWA for complex stock market prediction*," International Journal of Intelligent System, Wiley Online Library, Volume 37, 2022, PP.4586–4611

[21] P. Chhajer, M. Shah, and A. Kshirsagar," *The applications of artificial neural networks, support vector machines, and long–short term memory for stock market prediction*," Neural Computing and Application, Springer, Volume 2, https://doi.org/10.1016/j.dajour.2021.100015, 2022, PP. 1-12.

[22] D. Kumar, P. Sarangi, and R. Verma," *A systematic review of stock market prediction using machine learning and statistical techniques*," Materials Today: Proceeding, Elsevier, Volume 49, 2022, PP. 3187-3191.

[23] Qiu, Y., Song, Z. & Chen, Z. "*Short-term stock trends prediction based on sentiment analysis and machine learning,"* Soft Computing, Springer, Volume 26, https://doi.org/10.1007/s00500-021-06602-7, 2022, PP. 2209–2224.

[24] N. Das, B. Sadhukhan, T. Chatterjee, et al.," Effect of public sentiment on stock market movement prediction during the COVID-19 outbreak," Social Network Analysis and Mining, Springer, Volume 12, Issue 92, 2022, PP. 1-22.

[25] J. Wu, Z. Li, G. Srivastava, M. Tasi, and J. Lin," A graph-based convolutional neural network stock price prediction with leading indicators," Wiley Online Library, Volume 51, https://doi.org/10.1002/spe.2915, 2021, PP. 628-644.

[26] T. Aldhyani and A. Alzahrani," Framework for Predicting and Modeling Stock Market Prices Based on Deep Learning Algorithms," Electronics, MDPI, Volume 11, Issue 39, https://doi.org/10.3390/electronics11193149, 2022, PP. 1-19.

[27] M. Kumbure, Ch. Lohrmann, P. Luukka, et al.," Machine learning techniques and data for stock market forecasting: A literature review," Expert Systems with Applications, Elsevier, Volume 197, https://doi.org/10.1016/j.eswa.2022.116659, 2022, PP. 1-41.

[28] Q. Liu, Z. Tao, Y. Tse, et al.," Stock market prediction with deep learning: The case of China," Finance Research Letters, Elsevier, Volume 46, https://doi.org/10.1016/j.frl.2021.102209, 2022.

[29] Hochreiter, S.; Schmidhuber, J. Long Short-Term Memory Recognition. *Neural Comput.* **1997**, *9*, 1735–1780.

[30] Chen, Y.; Fang, R.; Liang, T.; Sha, Z.; Li, S.; Yi, Y.; Zhou, W.; Song, H. Stock Price Forecast Based on CNN-BiLSTM-ECA Model. *Sci. Program.* **2021**, *2021*, 2446543

[31] Yadava, A.; Jhaa, C.K.; Sharanb, A. Optimizing LSTM for time series prediction in Indian stock market. *Procedia Comput. Sci.* **2020**, *167*, 2091–2100

[32] Zaheer, S.; Anjum, N.; Hussain, S.; Algarni, A.D.; Iqbal, J.; Bourouis, S.; Ullah, S.S. A Multi Parameter Forecasting for Stock Time Series Data Using LSTM and Deep Learning Model. *Mathematics* **2023**, *11*, 590. https://doi.org/10.3390/math11030590

[33] LSTM. Available online: **https://www.analyticsvidhya.com/blog/2017/12/fundamentals-of-deep-learning-introduction-to-lstm/** (accessed on 17 February 2023).

6.3. Appendix.

Sample code:

- **Libraries.**

```
In [1]: import math as mth #math library
        import numpy as np # linear algebra
        import pandas as pd #data processing, CSV file I/O
        import pandas_datareader as web #importing dataset
        from sklearn.preprocessing import MinMaxScaler #Data preprocessing (Data Scaling)
        from sklearn.metrics import accuracy_score,recall_score

        import os
        os.environ['TF_CPP_MIN_LOG_LEVEL']='2'
        import tensorflow as tf

        from keras.models import Sequential,load_model
        from keras.layers import Bidirectional,LSTM,Dense,Dropout,BatchNormalization,GRU,SimpleRNN
        from keras import optimizers,regularizers
        from keras.callbacks import ModelCheckpoint,EarlyStopping


        #Data visualization
        import matplotlib.pyplot as plt
        %matplotlib inline
        import seaborn as sns



        import yfinance as yf
        from yahoo_fin import stock_info as si
        import yahoo_fin.stock_info as yfs
        from stockstats import StockDataFrame as Sdf
        from datetime import datetime, timedelta
        from requests.exceptions import ConnectionError
```

Fig.79. Libraries (Model Code).

- **Importing Dataset.**

```
In [2]:  path=path="/home/ahmed/Downloads/IBM.csv"
        df=pd.read_csv(path,index_col='time')
        df.head()
```

Out[2]:

time	open	high	low	close	volume
3/25/2022 19:19	131.300	131.30	131.300	131.30	718
3/25/2022 19:18	131.300	131.30	131.300	131.30	282
3/25/2022 17:50	131.300	131.30	131.300	131.30	100
3/25/2022 16:59	131.340	131.35	131.340	131.35	1427
3/25/2022 16:42	131.326	131.35	131.326	131.35	1326

```
In [3]: df.shape
Out[3]: (216883, 5)
```

Fig.80. Dataset &Data Shape (Model Code).

- **Data Scaling.**

```
In [7]: scaler=MinMaxScaler(feature_range=(0,1))
        data_scaled=scaler.fit_transform(data.values)

In [8]: data_scaled

Out[8]: array([[0.79987318],
               [0.79987318],
               [0.79987318],
               ...,
               [0.01728156],
               [0.03389616],
               [0.02318897]])
```

Fig.81. Data Scaling (Model Code).

- **Data splitting**

```
In [9]: training_data=data_scaled[0:training_length,:]

In [10]: x_train=[]
         y_train=[]

         for i in range(1,training_length):
             x_train.append(training_data[i-1:i,0])
             y_train.append(data_scaled[i,0])

         x_train,y_train=np.array(x_train),np.array(y_train)

In [11]: print(x_train.shape)
         print(y_train.shape)

         (173505, 1)
         (173505,)

In [12]: x_train=np.reshape(x_train,[x_train.shape[0],x_train.shape[1],1])
```

Fig.82. Data splitting (Model Code).

- **Model Structure.**

```
In [13]: # Build the BidirectionlaGRU & LSTM model
model = Sequential()

model.add(Bidirectional((GRU(100, return_sequences=True ,activation='tanh',activity_regularizer=regularizers.l1(0.001)
                        ,input_shape=(x_train.shape[1],1)))))
BatchNormalization()
model.add(Dropout(0.2))

model.add(LSTM(100, return_sequences=True ,activation='relu',activity_regularizer=regularizers.l1(0.001)))
BatchNormalization()
model.add(Dropout(0.2))

model.add(LSTM(50, return_sequences=True ,activation='relu',activity_regularizer=regularizers.l1(0.001)))
BatchNormalization()
model.add(Dropout(0.2))

model.add(LSTM(50, return_sequences=False ,activation='relu',activity_regularizer=regularizers.l1(0.001)))
BatchNormalization()
model.add(Dropout(0.2))

model.add(Dense(1,activation='linear'))

# Compile the model
optimizer=optimizers.RMSprop(learning_rate=0.0001)
model.compile(optimizer=optimizer, loss='mse',metrics=['mse', 'mae'])

#check_point=ModelCheckpoint('./Model_PricePrediction.h5',monitor="val_loss",verbose=2,save_best_only=True)
#Early_Stopping monitor
#early_stopping=EarlyStopping(patience=3)
# Train the model
num_epochs=35
history=model.fit(x_train, y_train, batch_size=32,epochs=num_epochs,shuffle=True,validation_split=0.3,
                  #callbacks=[early_stopping],
                 ) #traning_data 70% , validation_data 30%
```

Fig.83. Model structure and Regularization technique (Model Code).

- **Model evaluation.**

```
In [24]: from sklearn.metrics import mean_absolute_error, mean_squared_error, r2_score
print(f'R2: {r2_score(y_test, y_pred_test)}, \
MAE: {mean_absolute_error(y_test, y_pred_test)}, \
MSE: {mean_squared_error(y_test, y_pred_test)}')

R2: 0.9942822284249528, MAE: 0.20733145445123435, MSE: 0.09182932160021842

In [25]: print(f'R2: {r2_score(y_train, y_pred_train)}, \
MAE: {mean_absolute_error(y_train, y_pred_train)}, \
MSE: {mean_squared_error(y_train, y_pred_train)}')

R2: 0.9990886167992077, MAE: 0.004722249501962653, MSE: 4.50659622040718e-05
```

Fig.84. Model evaluation (Model Code).

```yaml

  # The following adds the Cupertino Icons font to your application.
  # Use with the CupertinoIcons class for iOS style icons.
  cupertino_icons: ^1.0.2
  google_nav_bar: ^5.0.6
  syncfusion_flutter_charts: ^19.1.54
  fl_chart: ^0.40.5
  animate_do: ^2.0.0
  carousel_slider: ^4.2.1
  google_fonts: ^2.1.0
  charts_flutter: ^0.10.0
  date_format: ^2.0.7
  smooth_page_indicator: ^1.0.0
  animated_splash_screen: ^1.3.0
  intl_phone_field: ^3.1.0
  lottie: ^1.2.1
  tflite_flutter: ^0.9.0
  cloud_firestore: ^4.5.0
```

Fig.85. flutter libraries.

```dart
@override
Widget build(BuildContext context) {
  return MaterialApp(
    debugShowCheckedModeBanner:false,
    title: 'GNav',
    theme: ThemeData(
      primaryColor: Colors.grey[800],
    ),  // ThemeData
    home:Intro(),
    routes: {
      'Start':(context) => Login(),
      's_back':(context) => Start(),
      's_back_wallet':(context) => firstpage(),
      'Buy':(context) => Buy(),
      'Register':(context) => Register(),
      'IBM':(context) => StocksPage(),
      'Login':(context) => Nav_Bar(),
    },
  );  // MaterialApp
```

Fig.86. main dart code.