Czech University of Life Sciences Prague

Faculty of Economics and Management

Department of Information Technology

Master's Thesis

Artificial Intelligence methods for decision making

Gaurav Keshavkumar Patel

CZECH UNIVERSITY OF LIFE SCIENCES PRAGUE

Faculty of Economics and Management

DIPLOMA THESIS ASSIGNMENT

Gaurav Keshavkumar Patel

Systems Engineering and Informatics
Informatics

Thesis Title

Artificial Intelligence methods for decision making

Objectives of thesis

Give an overview of Artificial Intelligence methods that are used to support decision making. Design a suitable solution method for the selected specific decision problem and implement it using publicly available so ware and data.

Methodology

The student will give an overview of Artificial Intelligence methods that are used to support decision making in Stock Market Prediction and Efficiency Analysis using Recurrent Neural Network and with Python. Then he will focus on solving a specific decision problem and use one of these methods to solve it. The proposed solution will be verified on data publicly available in a database intended for testing Artificial Intelligence methods. For testing the student will use freely available software.

The proposed extent of the thesis

60 pages

Keywords
Artificial intelligence, Neural networks, Machine learning, Decision making, Stock Prediction

Recommended information sources

Gupta I., Nagpal G.: Artificial Intelligence and Expert systems, 2020, Boston

Expected date of thesis defence
2021/22 WS – FEM

The Diploma Thesis Supervisor
doc. Ing. Arnošt Veselý, CSc.

Supervising department
Department of Information Engineering

Electronic approval: 11. 3. 2022		Electronic approval: 14.3. 2022
Ing. Martin Pelikán, Ph.D.	Prague on 30.03.2022	**Doc Ing. Tomáš Šubrt, Ph.D.**
Head of department		Dean

Declaration

I declare that I have worked on my diploma thesis titled "Artificial Intelligence methods for decision making" by myself and I have used only the sources mentioned at the end of the thesis. As the author of the diploma thesis, I declare that the thesis does not break copyrights of any their person.

In Prague on

Acknowledgement

Any individual task carried out cannot be completed without the help of different sources which directly or indirectly contribute to the achievement of the project.

I would like to thank my supervisor doc Ing. Arnost Vesely, Csc. support for his coaching and leadership in this study. As a consultant, he supported me with his unfailing support and advice during the project.

The facilities and services on campus and at the library helped to extend my understanding of corporate culture and business results. I was able to expand on my research by using the available case studies, research papers and documents.

I want to extend my appreciation to my classmates and batch colleagues for their efforts to help me appreciate and understand the complex views of individuals in society and to research consumer behaviour.

My close and dear friends' endless unconditional encouragement was a fantastic inspiration for success in this research during the project. Any Thanksgiving is incomplete without acknowledging my best contribution to family and friends.

Thank you all for your patience, understanding and support in carrying out this thesis.

“Artificial Intelligence methods for decision making”

Abstract

Artificial Intelligence methods are used for decision making in Stock Market Prediction and Efficiency Analysis using Recurrent Neural Network and with Python. Forecasting stock market prices have always been challenging task for many businesses analyst and researchers. In fact, stock market price prediction is an interesting area of research for investors. For successful investment, many investors are interested in knowing about the future situation of the market. Based on the study's key findings, the research with respect to the assessment and help to prediction regarding the use of stock markets has been found very promising using various artificial intelligence methods for decision making. In this project, we study the problem of stock market forecasting using Recurrent Neural Network (RNN) with Long Short-Term Memory (LSTM). The purpose of this project is to examine the feasibility and performance of LSTM in stock market forecasting. There are also some significant implications reported that need to be further analyzed in continuing the approach. The trading associated with stock markets has been found convincing and positive in recent times as different tools discussed in the methodology section were found effective in determining the efficiency and predictors of the stock market. Using Kaggle to collect the Google dataset was convincing, based on relevant results generated. Thus, the study aims to use a primary dataset throughout the project due to convincing and satisfying results. The major recommendation regarding this project was to contain the daily change in ratios of stock price by storing it in a CSV file. The storing of data in this file would be helpful in making the import and export of data easier in Python. The stock sector is one of the widely spread sectors in the world. The different functions included in the stock are bonds, stocks and other agreements between buyers and sellers. With the generation of promising results from the findings and analysis, there are some limitations or implications observed during the study. With their ability to discover patterns in nonlinear and chaotic systems, neural networks offer the ability to predict market directions more accurately than current techniques which helps the investor to make the better decision making in stock market.

Keywords

Artificial Intelligence, Neural networks, Machine learning, Decision Making, Stock Prediction

"Metody umělé inteligence pro rozhodování"

Abstraktní

Prognózování cen na akciovém trhu bylo vždy náročným úkolem pro mnoho obchodních analytiků a výzkumníků. Ve skutečnosti je predikce cen akciového trhu pro investory zajímavou oblastí výzkumu. Pro úspěšnou investici má mnoho investorů zájem vědět o budoucí situaci na trhu. Na základě klíčových zjištění studie se výzkum s ohledem na hodnocení a pomoc při predikci ohledně využití akciových trhů ukázal jako velmi slibný s využitím různých metod rozhodování. V tomto projektu studujeme problém prognózování akciového trhu pomocí rekurentní neuronové sítě (RNN) s dlouhou krátkodobou pamětí (LSTM). Účelem tohoto projektu je prozkoumat proveditelnost a výkonnost LSTM v prognózování akciového trhu. Uvádí se také některé významné důsledky, které je třeba dále analyzovat, aby bylo možné pokračovat v tomto přístupu. Obchodování spojené s akciovými trhy bylo v poslední době shledáno přesvědčivým a pozitivním, protože různé nástroje diskutované v sekci metodologie byly shledány účinnými při určování účinnosti a prediktorů akciového trhu. Použití Kaggle ke sběru datové sady Google bylo přesvědčivé na základě vygenerovaných relevantních výsledků. Studie si proto klade za cíl používat primární datový soubor v celém projektu kvůli přesvědčivým a uspokojivým výsledkům. Hlavním doporučením ohledně tohoto projektu bylo zahrnout denní změny v poměrech ceny akcií uložením do souboru CSV. Ukládání dat do tohoto souboru by pomohlo usnadnit import a export dat v Pythonu. Akciový sektor je jedním z nejrozšířenějších sektorů na světě. Různé funkce zahrnuté v akciích jsou dluhopisy, akcie a další dohody mezi kupujícími a prodávajícími. S generováním slibných výsledků ze zjištění a analýzy existují určitá omezení nebo důsledky pozorované během studie. Díky své schopnosti odhalit vzory v nelineárních a chaotických systémech nabízejí neuronové sítě schopnost předpovídat směry trhu přesněji než současné techniky, což pomáhá investorovi činit lepší rozhodování na akciovém trhu.

Klíčová slova

Umělá inteligence, Neuronové sítě, Strojové učení, Rozhodování, Předpověď akcií

Table of Contents

Chapter 01: Introduction

1.0 Introduction

With the use of Artificial methods in prediction has played an important role. Prediction can be seen as an important aspect of the financial market. The stock sector is considered to be one of the widely spread sectors in the world. The different functions which are included in the stock are bonds, stocks and different agreement between buyers and sellers. As the nature of stock price is dynamic hence the prediction is becoming difficult. Different techniques have been used to forecast the stocks' price and the trend which are going on in the market (Yoo, Kim and Jan 2005). But the utilization of ML and the ANN is one of the most efficient as well as interesting methods. The method of forecasting is considered to be an example of signal processing because of the high noise, non-stationary and non-linearity. The noise aspect shows the difference between the past trend in the stock and future price. This difference can be called the information gap. Different factors impact the stock price which typically includes physical, rational, political, irrational etc. These all factors make the prices fluctuate and make it relatively difficult to predict. Due to this the stock traders has shifted towards the use of intelligent systems (Sharma, Bhuria and Singh, 2017). Also with the advancements in technology different algorithms have been formulated which gives efficient results.

The application of neural networks in forecasting stock market prices. With their ability to discover patterns in nonlinear and chaotic systems, neural networks offer the ability to predict market directions more accurately than current techniques which Help the Investor and Traders to make a best decision to invest in Stock Market.

1.1 Background

The stock market is an important sector in today's world. It is used for trading the stock at an appropriate price. Two main aspects drive the stock which typically includes supply and demand. With the increased use of technology, people tend to use intelligent trading systems for the prediction of stock market trends. According to the study conducted by Vadlamudi (2017), the main objective of the trader is to forecast the price as fast as possible so that he can buy before the price gets high and sell before the price falls. So, for this purpose, different efficient algorithms have been proposed and used. According to the study fundamentals analysis has been previously used for prediction. Since it can involve more human mistakes so different methods have been developed to eradicate this probability.

According to the study given by Ahangar (2014), the stock market is considered to be non-parametric, noisy and non-linear. In the light of the study conducted by Oncharoen and Vateekul (2018), three different traditional approaches have been used for prediction. These approaches include traditional time series forecasting, technical analysis and machine learning techniques. Also, it can be seen from the prior studies different machine learning methods have been used which involves the classical regression method. These methods further include the polynomial and the linear regression. This regression successfully calculates the prediction in the form of graphs. Conventional models which incorporate the ARIMA, moving average and smoothing makes the forecasting linearly.

Furthermore, it can be studied that support vector machines and ANN are the most utilized methods for prediction. Artificial neural networks include the technical analysis of the data for forecasting, and it contains the different threshold functions. After assigning weights to the data, different training is applied to the data to make suitable predictions (Shadbolt and Taylor, 2002). Therefore, the above study uses a different method which is recurrent neural networks to minimize the probability of error and prediction of Indian stock prices that occurred by using the artificial neural networks.

1.2 Problem statement

In data analysis, prediction and modelling play a significant role. The stock market is the most popular sector in today' world. The use of conventional methods can contain more probability of mathematical mistakes. The purpose of prediction is to increase the investment rate and opportunity in the business. With the implementation of algorithms predictions about the stock can be made. Previously different artificial neural networks and convolution neural networks have been used but it has an error which is approximately 20% (Bing, Zhao and Hing, 2012). Considering this the model has been designed which uses the Recurrent neural network to forecast the price of the stock but with an occurrence of less percentage of error. Also based on the results efficiency can be checked which will determine the reliability of the model.

1.3 Research aims and objectives

The study aims to devise a model which can successfully make the prediction based on the data. Also, this study calculates the efficiency of the system as well. For this purpose, recurrent neural networks and jupytor will be used. The below-mentioned objectives are followed in the current study.

- Artificial Intelligence methods are used for decision making in Stock Market Prediction and Efficiency Analysis using Recurrent Neural Network and with Python
- To study the significance of machine learning techniques and their concepts in the stock market.
- To give an overview and importance of machine learning techniques and their models for the stock market.
- To evaluate the prior studies and literature over the concepts of machine learning and the various number of tasks for stock market prediction.
- To make the prediction as well as find the efficiency of the model by using the recurrent neural networks.

1.4 Research question

How are the machine learning techniques used to predict the data?

1.5 Motivation and Need

Below are the issues which lead to the need and motivation for using machine learning techniques for calculating retention:

1.5.1 Optimizing Decisions Instead of Predictions

ML techniques were implemented on larger data sets to predict outcomes as well as to improve the decision-making process. For example, in stock market prediction, it is important to predict the trend of the stock market but at the same time, it is also imperative to learn what actions would lead to changing the trends. If this need is fully met by the machine, the enterprise will reap great benefits using the predictions. Along with accurate predictions, decision making will be less risk induced with greater profits.

1.5.2 Scaling To Extremely Large Data Sets

The computation done by humans can only be done on limited datasets with less probability of being accurate all the time. This led to various problems which were solved by ML which had the ability and the capability to work on any data set irrespective of the size. Both simple and complex datasets can be used with higher accuracy, depending on the ML model. This also consumes lesser time which adds to another benefit.

1.5.3 Active Experimentation

Usually, the current data mining systems submissively accepted a fixed dataset. However, the need for new computer models that actively generated experiments optimally was needed to obtain relevant information for future learning. Therefore, the need for algorithms to optimize for the collection of the most informative. Data arose, keeping in mind both the expected benefits are risks that might take place.

1.5.4 Learning over Multiple Database and World Wide Web

The quantity of data along with its diversity is available on the internet and intranets is in huge quantity and continuously growing. Thus, methods of data mining might expand their access to this huge variety of data sets from various sources and increase their abilities to learn relevant regularities. To achieve this, new methods are needed to successfully extract information from the worldwide hypertext.

1.5.5 Prediction Accuracy through New Inventions

Another way to improve the accuracy of predictions is to invent new features which explain the data. This will pave new ways to locate the problems that arise in calculating accurate predictions and would also be updated to the current data needs and wants, maintaining consistency. In this time and age, the data sets must be updated to the current market holdings as the smallest of change can result in completely different predictions which will be used to make decisions for the entire enterprises.

1.6 Technology used

1.6.1 Jupyter Notebook

Jupyter notebook is an open-source application. It supports a language which is python and used to write files, equations, visualization and text.

Its uses include:

- Data cleaning
- Transformation
- Numerical solution
- Statistical modelling
- Data visualization
- Machine learning

Jupytor has its root that it assists Julia, python and R. Jupytor with the kernel, which is IPython, and it gives us the feasibility to write programs in Python. There are various Libraries which is used in the Project are as follows

- **NumPy**

  NumPy is python modules which provide scientific and higher-level mathematical abstractions wrapped in python. In most of the programming languages, we can't use mathematical abstractions such as f(x) as it would affect the semantics and the syntax of the code. But by using NumPy we can exploit such functions in our code. NumPy's array type augments the Python language with an efficient data structure used for numerical work, e.g., manipulating matrices. NumPy also provides basic numerical routines, such as tools for finding Eigenvectors.

- **Scikit Learn**

  Scikit-learn is a free software machine learning library for the Python programming language. It features various classification, regression and clustering algorithms including support vector machine, random forest, gradient boosting, k-means etc. It is mainly designed to interoperate with the Python numerical and scientific libraries NumPy and SciPy. Scikit-learn is largely written in Python, with some core algorithms written in Cython to achieve performance. Support vector machines are implemented by a Cython wrapper around LIBSVM.i.e., logistic regression and linear support vector machines by a similar wrapper around LIBLINEAR.

- **TensorFlow**

  TensorFlow is an open-source software library for numerical computation using data flow graphs. Nodes in the graph represent mathematical operations, while the graph edges represent the multidimensional data arrays (tensors) communicated between them.
  The flexible architecture allows you to deploy 6 computations to one or more CPUs or GPUs in a desktop, server, or mobile device with a single API.
  TensorFlow was originally developed by researchers and engineers working on the Google Brain Team within Google's Machine Intelligence research organization for the purposes of conducting machine learning and deep neural networks research, but the system is general enough to be applicable in a wide variety of other domains as well.
  TensorFlow is Google Brain's second-generation system. While the reference implementation runs on single devices, TensorFlow can run on multiple CPUs and GPUs (with optional CUDA and SYCL extensions for general-purpose computing on graphics processing units). TensorFlow is available on 64-bit Linux, macOS, Windows, and mobile computing platforms including Android and iOS.

- **Keras**

  Keras is a high-level neural networks API, written in Python and capable of running on top of TensorFlow, CNTK, or Theano. It was developed with a focus on enabling fast experimentation. Being able to go from idea to result with the least possible delay is key to doing good research. Keras allows for easy and fast prototyping (through user friendliness, modularity, and extensibility). Supports both convolutional networks and recurrent networks, as well as combinations of the two. Runs seamlessly on CPU and GPU. The library contains numerous implementations of commonly used neural network building blocks such as layers, objectives, activation functions, optimizers, and a host of tools to make working with image and text data easier. The code is hosted on GitHub, and community support forums include the GitHub issues page, a Gitter channel and a Slack channel.

- **Pandas**

  Pandas is a software library written for the Python programming language for data manipulation and analysis. In particular, it offers data structures and operations for manipulating numerical tables and time series. It is free software released under the license. The name is derived from the term "panel data", an econometrics term for data sets that include observations over multiple time periods for the same individuals.

Chapter 02: literature review

2.0 Introduction

The stock market is considered to be an important aspect of the industrial sector. People nowadays are more inclined towards the stock market for the sole purpose of profit. Due to the continuous advancements in technology different methods has been used to predict the changing trend and the patterns which have occurred in the stock market. This chapter comprises different sections which are gathered from studying and evaluating different prior studies. These studies discuss in detail the technologies, models, their respective results and findings.

2.1 Neural Network Models for Software Development Effort Estimation: A Comparative Study

As per this study which is conducted by Nassif et al., (2016), Machine learning techniques is used for the prediction of effort. It predicts the effort which is required for the development of the software. Software development is considered to be the most important feature of software project management. The manager needs to predict the effort which is employed or will be employed in the software development in a bidding process. This study proposed different models for this purpose which includes many machines learning models especially neural networks models. This study utilized four NN models which are multilayer perceptron, radial basis function neural networks, general regression and cascade correlation. These models are compared with each other based on predictive accuracy based on absolute error criteria, overestimate and underestimate the power of the model and classification of inputs. This study uses the ISBSG dataset for training and validating the models. This dataset is divided into five individual datasets based on productivity value. The inputs of these models are software size, the platform for development, resource level and language type. The result of this study is in the form of software effort. The results of this study show that these models overestimate 80% and the importance of every model differs from one another. Also, the results of this study show that cascade correlation neural network performs well as compared to other modes.

Another study conducted by Azzeh, Nassif and Banitaan (2018) states that Software development efforts estimation (SDEE) is considered to be important and is one of o the main tasks in software project management. It is considered important for the project manager to predict the efforts and costs in an efficient manner for the processes of bidding as the overestimation process will lead to bidding losses to occur and the process of underestimation would cause the loss of money in the project. There are several SSDE models that are used in machine learning; however, the most important models are machine learning models and neural network models are considered to the most prominent in the field of software project management. The study has used the models of Multiplayer perceptron model, General Regression Neural Network, Radial Basis function neural network and Cascade Correlation Model. The models are compared on the basis of predictive accuracy and overestimation and underestimation. The results of the study indicate that overestimation and underestimation are considered to the side effects of improper estimation and further compares the difference between the actual and estimated efforts. It has further been observed that the model of CNN outperforms other models in 60% of the data sets and the model of RBFNN ranked second and outperformed other models by 40%.

2.2 A Systematic Review of Machine Learning Techniques for Software Fault Prediction

According to this study conducted by Malhotra (2015), Software fault is predicted which is useful for the developer as well as the project manager. It is used by the software developers in different phases of the software development cycle for the identification of faulty modules and classes. There are different machine learning models and techniques which are used for this purpose. This study uses the systematic review for the analysis purpose. This study examines the performance capability of machine learning techniques to find the software fault prediction. Through the performance capability and efficiency of machine learning, different software fault prediction models have been formulated. Also, this study gives the comparison of performance accuracy of various machine learning techniques. The results of this study use the CK metrics as the input and the results are in the form of fault prediction rate. It is performed for the AIF version. The version which is used for AIF is 1.6. This version consists of 965 classes in total and out of which 777 classes has zero bugs. This contains the number of percentages of bugs and based on that percentage descriptive statistics and linear regression analysis has been done to find the fault prediction rate.

Another study conducted by Nti, Adekoya and Weyori (2020) indicates that the software defect prediction is considered to be an interest of research as it can predict the faults in the early stages and helps to improve the processes in early stages. Many researchers have contributed to the study and their efforts have caused an approach to be proposed that has helped in an effective and efficient manner. Different machine learning techniques have bene applied in the research to remove the fault data from the defect prone models, approaches, and frameworks and in the study over 40 Clarivate analytics have been used and detailed classification of machine learning techniques have been used for software data prediction. Many of the researchers have stated that the classifier that is used for the prediction is considered to important and it helps to identify the patterns of faults that might occur in the machine learning process and helps to cater risks in an efficient manner. Based on the results of the study NASA datasets have been used the metric of McCabe is the most widely used metric that is used to evaluate the accuracy of the models. This study examines the performance capability of machine learning techniques to find the software fault prediction.

2.3 Stock Market Prediction Using Text Mining

In the light of the study given by Nassirtoussi et al., (2014), the prediction model provides a great deal to gain profit and revenues. This paper reviews the techniques used to predict the trends and the patterns. This paper further reviews that two of the analysis is used. It is either the technical or the fundamental analysis. Technical analysis uses the price to predict the changes in the future whereas the fundamental analysis used the information from the reports, journals, and the news. This information is known as unstructured textual information. In the advanced technology world, where everything is available online this applies the use of text mining strategies to deduce the important information for the analysis of the ongoing behaviour of the market. This study uses text mining and NLP (natural language processing) for the prediction of market trends. The study assesses the sentiment polarity and sentiment emotions which is gained from different news and reports to forecast future movements. This study contains a large dataset gathered from 25 financial news. Also, they gathered the data from different 500 companies and uses the price as an indicator for future movements. This study has used two ML methods which typically includes the SVM and LSTM. The results of this study show sentiment analysis affected the price to fluctuate and also sentiment emotions improve the accuracy of the prediction. Further, this study shows that SVM performs better and produced accurate results due to the small size of the data set.

As per the study of Pejic, et al. (2019) market prediction is used to offer profits and avenues and is considered to be a stimulus for many researchers. Many of the researchers in this regard have used the method of technical analysis and has caused to focus on analyzing the direction of prices to predict the future prices and have caused relevant trends to be identified. This has caused valuable information of the market to be publicly available and this indicates the importance of text mining. Moreover, this has method of text mining has caused relevant strategies to be used for the analyzing the market and its impact on stock market prediction. The techniques have been reviewed and the methods of machine learning and deep learning have been compared for the sentiment analysis and has caused conclude the appropriate method that should be used for potential predictions. The results of this study show sentiment analysis affected the price to fluctuate and also sentiment emotions improve the accuracy of the prediction. The study further states that analyzing the historical prices is important for future predictions and the techniques will help to improve the prediction methods and enable future studies to increase their scope and predict the stock market in an efficient manner.

2.4 Stock Market Prediction Using Recurrent Neural Networks

According to this study given by Jahan (2018), the stock market is relatively difficult to predict. Various factors are responsible for the fluctuation of the prices and to determine the price. These factors include the trends of the market, the ratio of supply and demand, global economy, declaration of earning, history of price and sensitive financial information. The invention of different technologies helps to make the prediction comparatively easy and accurate. This study uses data mining and machine learning on the large size of data sets for the analysis and development. The prices of the stocks are predicted using the supervised machine learning algorithm. This study uses the RNN algorithm on time-series data. Also, the prediction that has been made is cross-checked with the actual price.

Another study conducted by Liu, et al. (2017) states that the forecasting of stock price returns is considered to be a challenge for the day traders to yield more returns and most of the studies have focused on machine learning in this regard. The machine learning techniques have been used on the algorithms to predict the returns on the stock. The recurrent neural network along with the long short-term memory (LSTM) has been used in the study to forecast the future returns. The study further states that it has the potential and the ability to keep memory of the historical stock returns to forecast the future stock return. The model of RNN and LSTM is used to store the recent information of stock prices and in the study dropouts of RNN layers are used to avoid the overfitting of the model. Therefore, to the accomplish the task of stock market prediction the stock returns have been calculated based on the closing price of the stock and these are considered as an input in the recurrent neural network. Moreover, this model is used to minimize the risk of error and the model is further compared with a feed forward neural artificial network.

2.5 Stock Market Prediction by Using Artificial Neural Network

In this study, which is given by Yetis, Kaplan and Jamshidi (2014), neural networks have been used to predict the patterns in the stock market with the help of a useful method of data mining which is artificial neural networks. This study evaluates that how the NASDAQ's stock is predicted with an artificial neural network, provided the number of inputs of market share. This study has used the real exchange rate value of NASDAQ. Also, this study utilized the feed-forward networks. This network was trained with the help of stock market price as an input. The input data is for 1 year (2012-2013). The results show that the model performs well for the prediction of the NASDAQ stock market.

Another study that is conducted by Radityo, Munajat and Budi (2017) states that prediction and analysis of the stock market data is playing a significant role in the global economy and has led to the overall forecasting models to improve through machine learning. The data storage and computation has become more advanced and has led to the overall forecasting techniques to improve. The process of the stock market is affected with uncertainty and different factors, and it has become an important process in the field of finance. There are many processes and algorithms that are used for the forecasting and predicting and the methods that are involved include non-linear auto regression and Auto regression moving average (ARMA) and these are considered as non-linear neural network models. These techniques are considered to be useful as they cause complex forms of data to be assessed in different models and enhances the process of deep learning. This helps to perform the tasks that are beyond the programming boundaries and due to high volumes of data being generated from the stock market, different patterns are used to make accurate predictions of the stock market. Furthermore, this study evaluates that how the NASDAQ's stock is predicted with an artificial neural network.

2.6 Time Series Forecasting Using a Hybrid Arima and Neural Network Model

According to this study given by Zhang (2003), time series prediction is one of the most useful techniques. Linear models are widely used for this purpose. ARIMA is considered to be the most appropriate model counted in linear models. It has been in use for three decades. According to the research related to different techniques and models, ANN and ARIMA have the highest precision in terms of performance and prediction. This paper has combined the methodologies of both ANN and ARIMA to make the hybrid methodology that is used in linear and non-linear modelling. The results obtained on the real data shows that the mixed model is more effective and improves the precision of prediction rather than using any single model thus emphasizing more on the hybrid methodology.

According to the study of Fathi (2019) the Discrete Wavelet Transform (DWT) has caused the surge in domains such as science and engineering to increase and the study the advantage of DWT is discussed for the time series forecasting. The study has stated that the novel technique of DWT is used for linear and non-linear data and the ARIMA and ARMA models have been used separately that used to predict and recognize the components and are used to improve the accuracy of the prediction. The hybrid method in the study is used for real time series data and results are further compared with the model of ARIMA and ARMA and the results indicates that the method of DWT is considered to be efficient and has provided predictions for each series. The results obtained on the real data shows that the mixed model is more effective and improves the precision of prediction rather than using any single model thus emphasizing more on the hybrid methodology. Moreover, there are several methodologies that have been used in study and the results of the study are different from each other. This has caused the models to be compared and have assisted to identify the reliable methods that can cause the accuracy of the predictions to increase.

2.7 Stock Market Prediction with Multiple Classifiers

According to this study given by Qian and Rasheed (2007), prediction of the stock market is considered to be appealing and difficult. This study demonstrates the hypothesis of the market that shows the prices of the stocks should increase gradually and should not be forecasted with an accuracy of more than 50%. This study further scrutinized the steadiness of the DJIA index to prove that all periods are not equally random. Also, the researcher has utilized the Hurst exponent to choose a period that has greater predictability than the rest of the periods. The parameter has been determined heuristically for formulating the patterns with the help of two methods which typically includes the auto-mutual data and false nearest neighbor methods. After the generation of patterns through these methods, various classifiers are trained. These classifiers include k-nearest neighbor, decision tree and ANN. The results of this study show that with the possible combination of the above model's prediction is achieved with an accuracy of 65%.

The study of Nabipour, et al. (2020) has indicated that stock market predictions are carried out through machine learning techniques and has aimed to develop efficient models that are able to provide accurate results. Several classifiers and ensembles have been applied in the study and through different combinations. However, there are issues associated with the ensembles that cause the data to be affected. The first concern is associated with the choice of regressor, and the classifiers and the second concern are associated with the combination of techniques that are used to assemble the regressors. The stock market is considered too volatile and fragile, and the price movement affects the consumers in a negative or positive manner depending on situation, therefore, different models are used to assess the movement in stocks and make predictions accordingly. The study indicates that the prices of the stocks are to be increased gradually and it should not be forecasted with an accuracy more than 50%. Moreover, there are different indexes that are used for the development of data through different time series, and it enables decisions to be made based on the results of the data and the model. The classifiers that have been used in the study includes Decision tree and ANN.

2.8Recurrent Neural Networks

This study which is given by Medsker and Jian (2001), shows that RNN has played an important role in research and development. Recurrent neural networks have been used to study sequential or time-varying patterns. A recurrent net is called a neural network with feedback means that it has close connections. Examples of the recurrent net involve the Hopfield, recurrent backpropagation, Boltzmann and BAM. According to this study, there are different applications of recurrent neural networks. Some of the applications include electric load forecasting, natural water flows forecasting, estimation of power etc. the architecture of the recurrent neural networks are relatively closed and connected. This can vary from application to application means that architecture can be partially connected. The variance of learning has produced several different techniques connected with the RNN. The main motive of the RNN is to improvise algorithms that are efficient and simple as well.

According to Raj and Ananthi (2019) recurrent neural networks (RNN) are used for the development of algorithms and sequential data and the first algorithm is used for internal memory and storage of data. These are considered to be powerful and belong to the most promising algorithms. They are mostly used in deep learning as they are relatively old, however, they have potential and provide accurate predictions based on the data that is available. The invention of different technologies helps to make the prediction comparatively easy and accurate. This study uses data mining and machine learning on the large size of data sets for the analysis and development. The prices of the stocks are predicted using the supervised machine learning algorithm. The stock sector is considered to be one of the widely spread sectors in the world. The different functions which are included in the stock are bonds, stocks and different agreement between buyers and sellers. As the nature of stock price is dynamic hence the prediction is becoming difficult. Different techniques have been used to forecast the stocks' price and the trend which are going on in the market. Therefore, the method of RNN is considered to be important for the development of data.

2.9 NSE Stock Market Prediction Using Deep Learning Models

According to this study which is given by Hiransha et al., (2018), the neural networks is considered to be one of the useful data mining techniques. This study says that it is the most used method by the analyst in the last 10 years. The economy is greatly influenced by the forecasting method. For this purpose, different algorithms are used. These algorithms are divided into linear and non-linear models. Linear models are ARIMA, AR, ARMA and MA whereas the non-linear models ARCH, GARCH, and NN. Also, this model utilizes various architectures of four different types is Multilayer perceptron (MLP), Long Short-term Memory, Recurrent Neural Networks and Convolutional Neural Networks. These models forecast the stock price based on the historical prices which are stored in the databases. Through investigation of the study, it can be found that two stock markets have been used which are the National stock exchange of India and the New York Stock Exchange. However, the training of the network was done with only one stock market which is the National Stock Exchange of India, and it was used to predict for five various companies with the help of both stock markets. Also, the convolutional neural network was observed to perform better than the rest of the models. Due to the common inner dynamics, the model was able to predict for the NYSE even though it was trained for NSE. The results of this study show that RNN performs better than ARIMA.

According to the study of Hiransha, et al. (2018) the prediction of stock market movement is considered to be significant in the field of financial studies and deep learning models have been considered to be important for the predictions. The study has presented a deep learning network to predict the movement and the data that has been collected is based on time series and it is further combined with the convolution neural network. The CNN is used for the extraction and the LSTM is used for the prediction of prices. The three-dimensional CNN is used for the input of data and technical indicators such as correlation and regression are used to assess the trends in the prices of stocks. The results of the study indicate that the framework outperforms the state of art models in predicting the stock price movement direction. These models forecast the stock price based on the historical prices which are stored in the databases. However, the training of the network was done with only one stock market which is the National Stock Exchange of India, and it was used to predict for five various companies with the help of both stock markets. Also, the convolutional neural network was observed to perform better than the rest of the models.

2.10 Bayesian Recurrent Neural Network Models for Forecasting and Quantifying Uncertainty in Spatial-Temporal Data

As per this study which is given by McDermott (2019), RNN is considered to be the non-linear model which is used in machine learning to represent the sequential relationship between the variables. Also, with the advancements in machine learning, Recurrent Neural Networks are used to predict complex systems too. One of the most common examples includes the dynamic Spatio-temporal method. In this study, uncertainty is quantified with the help of the Bayesian RNN model for nonlinear Spatio-temporal prediction. The existing model is applied to the Lorenz simulation as well. Also, these models were highly feasible with varying degrees of nonlinearity.

Another study of McDermott and Wikle (2019) indicates that RNN are non-linear dynamic tools that are most commonly used in the machine learning, and it supports the dynamics of machine learning. The model of Bayesian supports the increased complicated systems and enables the relationship between the variables to be identified. The study further states that saptio-temporal processes are used to represent the class of complex systems that can help to benefit with different models that are further compared based on their efficiency. Moreover, it is used to identify any uncertainties and supports the prediction of the stock market.

2.11 Long Short-Term Memory Recurrent Neural Network (Lstm-Rnn) Based Workload Forecasting Model for Cloud Datacenters

This study has emphasized the importance of cloud computing by stating that cloud computing is considered to be power consuming. Therefore, it tends to be expensive and extremely large scaling. In this study, the prediction of workload has been calculated by considering the issues and problems that occurred in the cloud data center. The results are obtained with the help of long short-term memory (LSTM) networks. LSTM is tested using three datasets of web server logs. The above model predicted the results in achieving high accuracy by minimizing the mean squared error up to 3.17×10^{-13}. The output of the model is stored in the resource manager which takes the current state of the datacenter. This model used the python language with Keras library which acts as a tool for implementing the model (Kumar, Goomer and Singh, 2018). Three datasets that have been used is HTTP traces of the NASA server, Calgary server and Saskatchewan server. The window size used for prediction is 5, 10,15,20,30, and 60 minutes.

2.12 Stock Trend Prediction Using Simple Moving Average Supported by News Classification

The simple moving average is one of many time series analysis technique. Time series analysis is a method of timely structured data processing to find statistics or important characteristics for many reasons. The simple moving average shows stock trend by calculating the average value of stock prices on specific duration. The prices that are used are closing prices at the end of the day. This technique can avoid noises and therefore smooth the trend movement. The main objective of financial news classification is to classify and calculate each news' sentiment value. The positive news is marked by sentiment value 11 which is greater than 0, while negative news is marked by less than 0 sentiment value. If there are news having 0 sentiment value, they will be omitted as their neutralism does not affect the stock trend. Machine learning using artificial neural network algorithm is used to predict a stock trend. The artificial neural network uses three features along with one label. The three features are a simple moving average distance which is a subtraction of long-term and short-term simple moving average, the total value of positive sentiment value for one-day news, and the total value of negative sentiment value for one-day news. Stock trend label is used and classified as uptrend and downtrend (Stefan Lauren Dra. Harlili S 2014). On one hand, learning component is done by a background process. On the other hand, prediction component is foreground process which is seen and interact with the user.

Chapter 03: Methodology

3.1 Research Philosophy and Approach

The research philosophy indicates the belief about the method used to collect, analyze, and use the data for the study. It provides the roadway towards the selection of the methods and techniques for conducting the study. The researchers are using three types of research philosophies. These include positivism, interpretivism and pragmatism. The positivism philosophy shows that knowledge can be derived from scientific techniques. This philosophy implies the study by using systematic and scientific approaches to provide better and credible research findings. Interpretivism is the philosophy that deals with the human interest in the study. It involves the principle that the researcher has the specified role to perform for observing the social world. Pragmatism is the philosophy which implies the application of two different techniques which are direct and critical in nature (Bell, Brymann and Harley, 2018). The current research uses the positivism philosophy by developing the study on the scientific methods since it is most suitable for the current study. The study is based on the stock market prediction, which is a scientific topic and can be better researched through the positivism philosophy. This philosophy is also considered suitable in this research as it derives the findings that are more credible and valid. Hence, this study has opted to choose this research philosophy for the current study.

The research approach, on the other hand, is the plan and procedure, which includes the steps of the broader assumption to the detailed data collection and analysis techniques. There are two research approaches used in the study, which includes inductive and deductive approaches. The main distinction between the two methods is the time frame of developing the hypothesis. The deductive approach forms the hypothesis at the start of the study, while the inductive approach develops the hypothesis at the end of the study (Bougie and Sekeran, 2016). In this study, the research approach selected is deductive since the study is based on the hypothesis that is being tested. The study has developed the assumptions based on the previous studies and theories and conducts the study to further increase this prospect's scope. Hence, the deductive approach is considered to be more suitable.

3.1 Research Design

Research design provides a suitable structure for the study. A research design's role is to guarantee that the data collected allows researchers to answer the question as rationally and unambiguously as possible (Marczyk 2021). A research design serves as a framework for the techniques and processes that will be used to perform the study. The design allows researchers to concentrate on acceptable research techniques for the study topic and set their studies up for success. Researcher's implement two main types of research designs: quantitative and qualitative, and in some rare cases where the study demands the use of both these designs, which is known as the mixed methods research design. The research design is based on the nature of the study and the data being collected to conduct the study (Sileyew 2019). Qualitative research finds correlations between gathered information and observations Statistical techniques can be used to validate or disprove ideas about naturally occurring events. Qualitative research techniques are used by researchers to understand "why" a certain theory exists as well as "what" respondents believe about it.

On the other hand, Quantitative research is used when statistical findings are required to obtain actionable information (Geoffrey 2019). Numbers give a more accurate view while making crucial business decisions. As a result, quantitative research techniques are essential for the advancement of any business. Insights obtained from precise numerical data and analysis are usually extremely accurate. beneficial when making business-related decisions in the future. For mixed methods design, both qualitative and quantitative techniques are used to conduct the research.

3.2 Data Collection Technique

The process of obtaining data from all relevant sources in order to solve the research problem, test the hypothesis, and evaluate the outcomes is known as data collection. The correct data must be collected to support the research argument and answer the research questions and hypothesis (Paradis 2016). There are two main types of data collection methods: secondary data collection methods and primary data collection methods. Secondary data collection methods consist of collecting data that is already existing in published books, journals, articles, and papers (Aborisade 2013). This data is old and has been used by previous researchers in their studies.

On the other hand, primary data is relatively new and is collected from surveys and interviews, mainly comprised of human experiences and interpretations of the research problem. For the following study, data is collected from secondary sources because of the nature of the research topic. Finally, data will be collected from pre-existing research to support the research argument and draw a meaningful conclusion for this study.

3.3 Data Analysis Techniques

3.3.1 Technical Analysis Methods

The technical analysis method is applied for forecasting the movement in the prices of the different securities or tradable instruments that are mostly subjected to the supply and demand forces such as stocks, futures, bonds, and exchange rates. This type of analysis in actuality might be seen to be the simple supply and demand forces study as reflected in the movements of market prices of the particular security. The technique is primarily used in the prices changes, and most of the analysts additionally track the numbers as well other than the prices such as figures of the open interest rate and the trading volume.

There have been many technical indicators that have been developed over the years by most of the analysts for attempting to predict the movements accurately in the future prices. The focus of most of the indicators are on the evlaution of the current trends of market that involves the areas of resistance and support. In constrast, there are some other indicators that are focused on determining strength of trends and the likelihood continuation. Some of the techniques also involve the trend lines, moving averages, momentum indicators just as the moving average convergence divergence indicators.

Technical analysts tend to use the technical indicators to different timeframes charts. The traders over the short term may apply charts that are ranging from the frames of the minutes to the hourly. On the other hand, the traders interested in evaluating the long-term movement in the prices scrutinise the daily, weekly or monthly data. Therefore, it is used in predicting the stock market prices for deriving the better estimation and forecasting.

3.3.2 Fundamental Analysis Techniques

Fundamental analysis techniques further apply accurate and public data for evaluating the value of security. There are although many analysts that are using the fundamental analysis for the valuation of the stocks, this valuation technique can also be applied to the different other securities as well. The investor might apply the fundamental analysis on the bond value by considering over the economic indicators such as interest rates and the overall state of the economy. The investors might also focus on the information that are related with the bond issues such as the shifts of the credit ratings of the issuers of bonds.

For the instruments of the stocks and equity, the technique applies the revenues, earnings, future growth, return on equity and the profit margin rate and other information for the underlying organizational value and the potential of the future growth as well. Regarding the stocks, the fundamental analysis is focused on the organization's financial statement, which is being evaluated. The stock markets are predicted under this technique in a proper and better way. It has also been identified as the efficient method for deriving the predictions for the stock market data of the particular stocks. Hence, the stock market is being predicted using this technique widely.

3.3.3 Traditional Time Series Prediction

Time series analysis is quite helpful to look at how the given asset or the security is changing over time. This technique is also used for evaluating the changes that are linked with the particular data point when compared to the shifts in the other variables over similar period. The example includes analyzing the daily closing prices of stocks for the provided stock in one year. The closing prices lists would be obtained from each day for the past year and listing them in chronological order. It can be a one-year daily closing price of the stock time series.

The analysts are interested to know if the time series of the stock represents the seasonality for evaluating the fluctuations at the regular times every year. The analysis in this aspect might need to consider the observed prices and relating them to the selected season. It might involve the traditional calendar seasons like winter or summer or any retail seasons like the holidays. Alternatively, the analysts might also record the share prices of the stocks changes since it links to the economic variable like unemployment rate. Through the correlation in the data points with the information linked to the selected economic factors, it can be observed that the patterns in the situation showing dependency between the data points and the variables selected. The traditional time series techniques for the evaluation and prediction of the stock market have been considered an essential technique that is more detailed and comprehensive. It provides a more efficient and effective analysis of the data series.

3.3.4 Machine Learning Methods

In the recent times, Machine Learning (ML) has been used in most of the research fields especially in the fields of economics and finance. Most of the researchers have used machine learning algorithms for creating the tools for evaluating the historical financial data and the other information related to aid in investment decision making. The study of Jeong et al. (2018) has used a machine learning algorithm for supporting the stock investment decision making by the use of the data of the financial news and social media. At the same time, the study of Cho and Nguyen (2018) has predicted the stock prices of the construction companies by using the non-linear prediction technique. However, machine learning techniques have been identified as more suitable for predicting the stock market prices.

The use of the time series or the historical financial data and the careful selection of the suitable models, data and features are important for producing the results that are more accurate and valid. The accuracy of the results is dependent on the efficient infrastructure, gathering related information and employing the algorithms (Alpaydin, 2014). It has been said that the ML results would be more accurate when the quality of the data is much better.

With the great success in the machine learning over the last years, it has transformed the way in which the investors are using the information, and it provides the analytic opportunities for the different kinds of investment. Hence, machine learning has been considered as an essential tool for helping the financial investment. The stock market prediction can be accurately and reliably managed through the application of these techniques. These techniques involve the prediction, clustering, classification, and others. They are considered to be the more accurate and effective techniques in the prediction of the stock market changes. However, it also has a certain level of limitations linked with the prediction of the securities. Obthong et al. (2020) has explained that it does not fit well with the non-linear time series for the variables. The model has been determined for one series and would not be suitable for any other series. It has also been said to be the time taking technique when the data is too large.

3.3.5 Deep Learning

The function of the artificial intelligence acts like the brain of the human for the data processing and for developing the patterns that are being applied for making the effective decisions. Deep learning has been one of the subjects in the artificial intelligence for the machine learning concepts that has the networks able to learn unsupervised from the data which is unstructured and unlabeled. It is also called as the Deep Neural Network or the Deep Neural Learning. Deep learning has been used across different industries for various activities and tasks. Commercial applications using image recognition, open-source platforms and the medical research tools exploring the probabilities of drug reuse for new ailments are examples of deep learning. It is also widely used to predict the stock market movements and the movements in the other securities as well. Researchers are widely using different sub types of machine learning for the prediction of the stock market movements. However, this study has applied the Recurrent Neural Network (RNN) to better predict the movements in the Indian stock market.

RNN is related with the neural network and has been better at the processing and modelling of the sequential type of data. The expression specified is that the RNN is able to memories the state of the past which is capable to be used for calculating the current state. There are many layers hidden that are non-independent and the current layer hidden input includes the input layer output and the previous hidden layer output. It is the main reason that RNN has been performing better to deal with the data which is sequential in nature. The benefit of using RNN is that it focuses on the context of data in the training process that is very much suitable for the stocks and Forex scenario since the fluctuation at the particular time might involve some links with the last trends. The study of Hu, Zhao and Khushi (2020) has considered it to be the most effective technique for the prediction of the stock market prices. It has implied that this technique provides with the more accurate and suitable analysis of the stock market prediction and the results obtained are more accurate and robust. Hence this study has considered this technique for the prediction of the stock market using the Python for the better evaluation of the trends in the stock market. It can provide with the more enhanced analysis for increasing the credibility of this study and hence the researcher has considered it to be more suitable for the current study.

CHAPTER 04: Practical Part

4.0 Introduction:

This chapter is one of crucial components of the current study that presents the main structure, tools and techniques which is used to provide evidence in the support of core points of the research. Moreover, it helps to assess the main idea of the tool, components and its significance to the topic which aids in discovering the potentialities and drawing conclusions from it. Besides, the system architecture of the study shows the main steps which is used to construct the system in an efficient way. It also sheds light on the designing of the model, main model which helps to predict the stock market of India. Further it gives an insight of the several libraries which are used in the current study. This chapter highlights the key points which summarize the security and privacy challenges and how the implementation of various algorithms of machine learning is practiced in order to provide the useful results and outcomes.

4.1 Prediction Model:

Sequential prediction is the most observed problem over a long time ago. They are referred to as the hardest problem which has been occurred in the data science industry. Sequence predictions has been used widely to predict the number of patterns which includes the finding of patterns in stock market data, obtaining the movie plots to obtain the success or the failure rate of the movies in the market, predicting the translation of language to forecast the next word which can appear on the keyboard. Traditional methods were obtained in such a way that it operates and predicts on the data which is previously obtained and on the basis of prior understanding of the model. As per the study of Sun et al., (2017), the traditional neural networks were not smart enough to predict the variable on the basis of reasoning which is related to the events or the data. Recurrent neural networks solved the above problem by storing the information in forms of loops thus it allows the pertaining of the data.

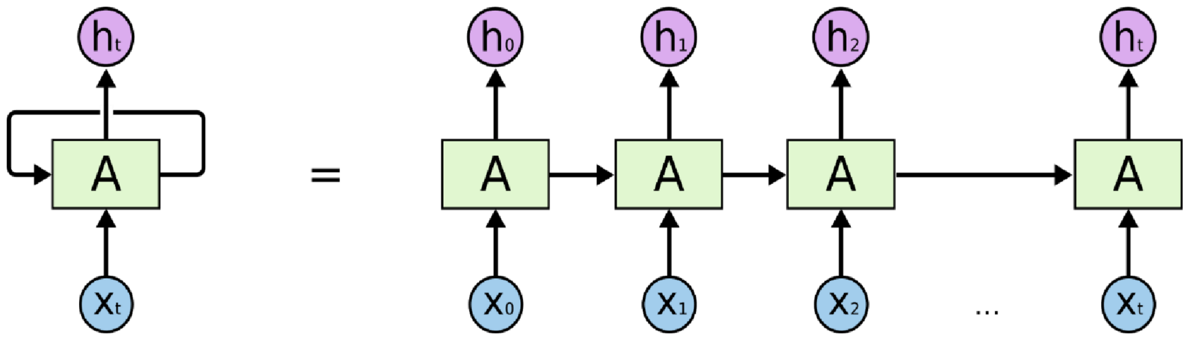


Figure 1(Recurrent neural networks having hoops)

RNN is like multiple copies of the same network where each copy passes information to the other. The list Like structure makes them appropriate for sequential data (e.g. time series, natural language, etc.) Thus, the diagram gives the proper explaining of the data which is stored in the form of loops. Chunk of neural networks has been stored in A, then it is dependent on the input x to give the result in h. thus the value rotates in the loop which is passed from one step to the next. Another important point which serves as the reason behind using the RNN has the ability to provide the several copies of the same network hence acting as a messenger to the output variable. RNN has been used in several problems which includes the speech recognition, modelling of language, translation, captioning of image and many more (Chawla et al., 2018). RNNs can handle a large amount of data and they can also handle the data with the long-term dependencies. Unlike humans, RNNs are not capable of learning through the model. So, for this purpose LSTM is used which can learn from the datasets.

LSTMs are considered to be the part of recurrent neural networks which can learn from the long-term data hence they are capable of learning long-term dependencies. All the RNNs acts in the form of chain which gives the idea of repeating information in the neural networks. LSTM also have the chain structure but the values or the structure which is used for repetition is different from the RNNs. There is total four layers which are interacting with each other instead of only one which is used in the RNNs. LSTM cell is held responsible for memorizing the values over different span of time. Long- and short-term memory cell has its own memory, and they have the ability to act like a human brain in making decisions (Bukhari et al., 2020). Cells which are part of the function of various inputs obtained from the prior steps are termed as memory cells. Since the stock market includes the huge bulk of data, which is comparatively complex to process, the gradients of the system may become small as compared to the weight matrix and ultimately results in degradation of the learning rate of the system. It can also result in the loss of gradient. LSTM contains different components which contains the output gate, input gate and a forget gate. The cell of the LSTM memorizes the value for the long term, and it helps the gates to regulate the value.

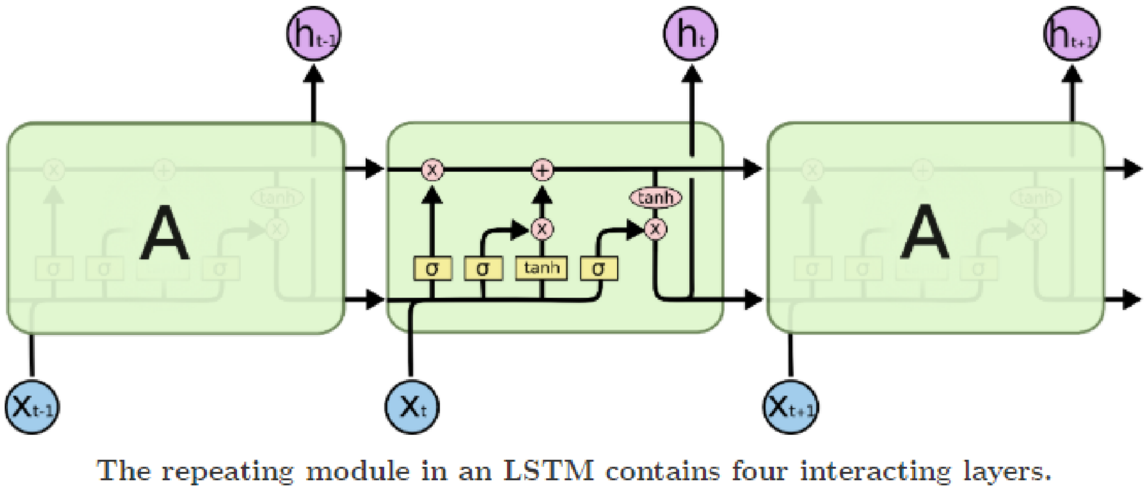
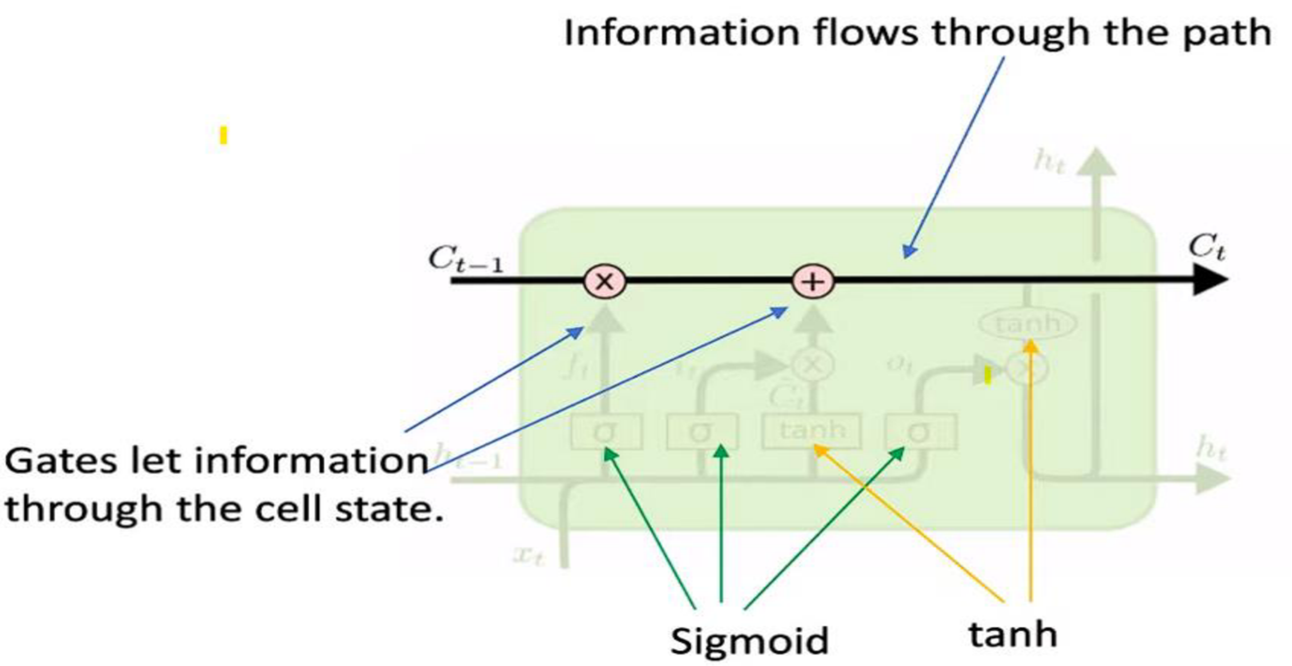


Figure 2(LSTM architecture)

Unrolling the LSTM:

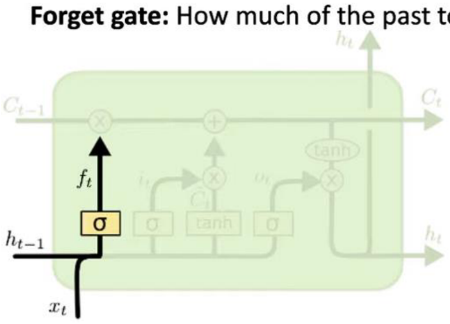


(Cell State)

Sigmoid: Sigmoid can output 0 to 1, it can be used to forget or remember the information.

Tanh: To overcome the vanishing gradient Problem tanh's second derivative can sustain for long range before going to Zero.

Forget gate: How much of the past to forget.

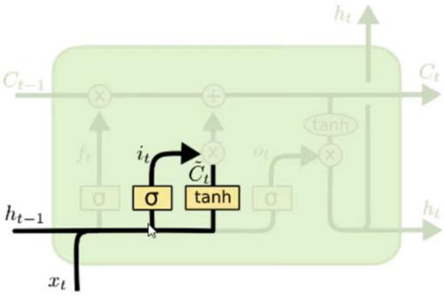


$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] \; + \; b_f\right)$$

Outputs a number between 0 and 1 for each number in the cell state.
0 to completely forget and 1 to keep all information.

(Forget Gate)

Input gate: What new information will be stored in the cell state.



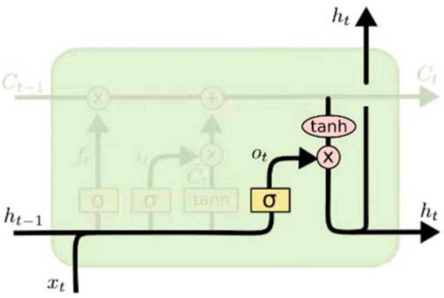
Sigmoid layer decides which values are updated.

$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] \; + \; b_i\right)$$
$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] \; + \; b_C)$$

tanh layer gives weights to the values to be added to the state.

(Input Gate)

Output gate: Decide what part of current cell makes to the output



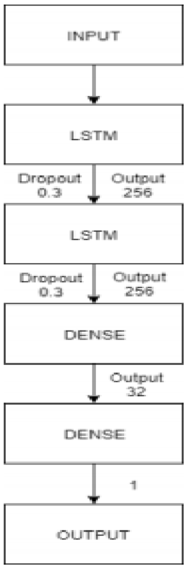
Sigmoid layer decides which part of cell state is selected for output.

$$o_t = \sigma\left(W_o \; [h_{t-1}, x_t] \; + \; b_o\right)$$
$$h_t = o_t * \tanh\left(C_t\right)$$

tanh layer gives weights to the values (-1 to 1).

(Output Gate)

(Workflow diagram)



Values used for prediction:

As the dataset was splitted in into training, testing and validation halves. The values used for prediction is stored in x_test variable. The values of the dataset are first transformed using min_max_scaler.fit_transform has also written in the code and then is divided among discussed halves. The screenshot below shows the values stored in x_test and used for prediction.

```
[8] print(x_train)

[[[0.9955265  1.         0.9989336  0.99750162]
  [0.99646145 0.99742512 0.99764384 1.        ]
  [0.99404666 0.99609512 0.99798402 0.99543496]
  ...
  [0.97106371 0.96891037 0.97285415 0.97796345]
  [0.9845045  0.97801096 0.98125914 0.98721639]
  [0.97533455 0.98018261 0.97827352 0.97799339]]

 [[0.99646145 0.99742512 0.99764384 1.        ]
  [0.99404666 0.99609512 0.99798402 0.99543496]
  [0.98836758 0.9894862  0.99390619 0.99028923]
  ...
  [0.9845045  0.97801096 0.98125914 0.98721639]
  [0.97533455 0.98018261 0.97827352 0.97799339]
  [0.96619945 0.97293695 0.97081637 0.97060104]]

 [[0.99404666 0.99609512 0.99798402 0.99543496]
  [0.98836758 0.9894862  0.99390619 0.99028923]
  [0.99372784 0.98102461 0.98488815 0.99113119]
```

Selection criteria of used dataset:

BSE Sensex Index Historical Price data has been selected which is an Indian index that broadly represents BSE and the market prices.

Formula applied for Recurrent Neural Network:

Simple RNN Formula:

$$h_t = Wf(h_t - 1) + W^{(hx)} x_{\{t\}}$$
$$y_t = W^{(S)} f(h_t)$$

Each error at Time (t):

$$\frac{\partial E}{\partial W} = \sum_{t=1}^{T} \frac{\partial E}{\partial W}$$

Chain Rule:

$$\frac{\partial E_t}{\partial W} = \sum_{t=1}^{T} \frac{\partial E_t}{\partial y_t} \frac{\partial y_t}{\partial h_t} \frac{\partial h_t}{\partial h_k} \frac{\partial h_k}{\partial W}$$

4.2 Network Architecture:

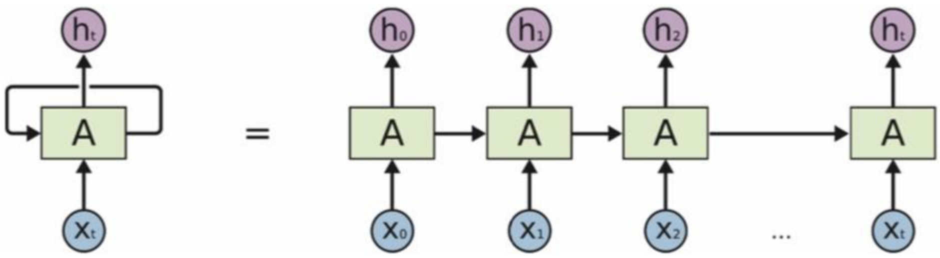
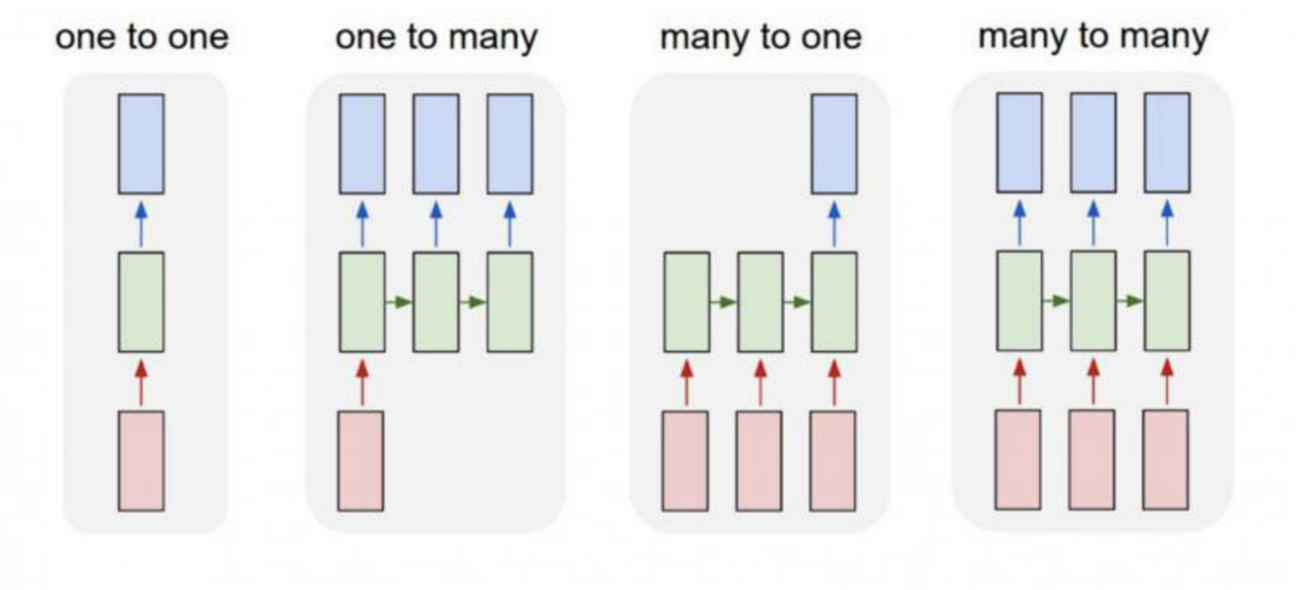


Figure 3(Network Architecture)

Explanation:

The RNN (Recurrent Neural Network) is an AI neutral network which is mostly used in NLP and speech recognition etc... RNN acknowledge data's array characteristics and utilize design to predict next possible structure/outline. RNN has 3 layers that they work on Input layer, Hidden layer and Output layer. There are 4 types present in RNN and they are as follows:



(RNN Layers)

1. **One to One:** fixed sized input to fixed sized output (e.g., Image Classification).
2. **One to Many:** Sequence output (e.g., Image captioning takes an image and outputs a sentence of words).
3. **Many to One:** Sequence input (e.g., sentimental analysis where a given sentence is classified as expressing positive or negative sentiment).
4. **Many to Many:** Sequence input and sequence output & Synced sequence input and output (e.g., Machine translation: an RNN reads a sentence in English and then output a sentence in Czech, Video Classification where we wish to label each frame of the Video).

The above diagram (3) shows the system architecture which is used. It shows the various methods or the steps which is used to conduct the project and is explained below.

4.2.1 Collection of datasets:

Collection of dataset implies the gathering of data for the purpose of testing. Data collection is primarily the most important part in the researcher. As per the study of Khalifa et al., (2021), data collection does not depend on the type of study as it ensures the information is reliable and valuable. Data can be collected with the help of any tool, and it can be collected from some secondary datasets. It is collected for the purpose of analysis upon which the decisions are made on the data collection. In our study, our dataset is in the form of CSV file. The dataset that is obtained consist of the daily BSE Sensex Index Historical prices.

It has been studied that Kaggle provides the dataset which is referred to as the financial and the economic data with the lot of auto-generated datasets (Chiang, Chiu and Kuo,2021). Further Kaggle gives the liability to provide the small python library which is used to access the data pragmatically. Also, the library is useful for the calculation of daily changes which occurs in the stock price. Hence the data is the daily basis data. Data can be stored in various forms but in this particular study dataset is stored in the form of CSV. The biggest advantage of storing the data in CSV is that it provides an easy handling of data and also it is easy to import and export the dataset with the help of Python using Panda's library.

4.2.2 Import the dataset

After collection of the dataset, it is first split into the test and the training of the datasets. After successful splitting of data, the dataset is given some unique and separate IDs to the variables which are used in the study. Further the data related attributes include the data of the stock, and the amount is also presented with each of the data variable corresponding to the date. The trained dataset is imported using the different libraries such as the pandas, NumPy and many more relevant to the procedure and they should be made to simplify the analysis techniques. The data is imported with the various steps that includes the load the data, creating labels, automated feature engineering, and machine learning.

```
# import all stock prices
df = pd.read_excel("/content/BSE Sensex Index Historical Prices.xlsx", index_col = 0)
df.info()
df.head()

# number of different stocks
# print('\nnumber of different stocks: ', len(list(set(df.symbol))))
# print(list(set(df.symbol))[:10])
```

Figure 4(importing the csv file)

4.2.3 Loading the dataset:

The first step which is done is the loading of data, it implies the loading of only one portion of the dataset to create an easy accessibility of the data. Data is first present in the unstructured form and then it can be loaded by partitioning of the datasets. After loading the datasets, pipeline is run on every partition to generate the final model. Data is loaded with the help of feature tools which is present in the python language.

Date	open	close	low	high
2021-10-12	60045.75	60284.31	59885.39	60331.74
2021-10-11	60099.68	60135.78	59811.42	60476.13
2021-10-08	59960.39	60059.06	59830.93	60212.30
2021-10-07	59632.81	59677.83	59597.06	59914.91
2021-10-06	59942.00	59189.73	59079.86	59963.57

Figure 5(loading the dataset in columns)

4.2.4 Create labels:

After loading the datasets, labels are inferred on the data which is considered to be the necessary part in supervised machine learning. The model helps to explain the predictive model which is employed to forecast the Indian stock price. After creating labels training is done on the datasets.

4.2.5 Automated feature engineering:

With the labels in hand, deep feature synthesis is used to automatically generate features. During deep feature synthesis, different variables has been specified. It is deemed prerequisite to scale the values and the data so that all the values which is present in the data falls into some specific range. For this purpose, Min-Max Scalar is used in the process. In other words, min-max scalar is the process which helps to normalize the input and output values of the dataset. In this way, the values come in range of [0, 1] which means that the minimum value is 0 and the maximum value of the variable is going to be 1. For the purpose of normalization of the dataset, fit and transform is used to train the data and transform is used for testing the data because the scale is a known figure in the testing of data. Hence the normalization method implies the value of the variable will become 0 and 1. Also to get the data of India, explicitly India has been applied in the Deep feature synthesis method.

```python
plt.figure(figsize=(25, 10));
plt.subplot(1,2,1);
df.columns = ['open', 'high', 'low', 'close']
plt.plot(df['open'].values, color='blue', label='open')
plt.plot(df['close'].values, color='green', label='close')
plt.plot(df['low'].values, color='black', label='low')
plt.plot(df['high'].values, color='red', label='high')
plt.title('stock price')
plt.xlabel('time [days]')
plt.ylabel('price')
plt.legend(loc='best')
```

Figure 6(Plotting price and volume of DF)

```python
def normalize_data(df):
    min_max_scaler = sklearn.preprocessing.MinMaxScaler()
    df['open'] = min_max_scaler.fit_transform(df.open.values.reshape(-1,1))
    df['high'] = min_max_scaler.fit_transform(df.high.values.reshape(-1,1))
    df['low'] = min_max_scaler.fit_transform(df.low.values.reshape(-1,1))
    df['close'] = min_max_scaler.fit_transform(df['close'].values.reshape(-1,1))
    return df

# function to create train, validation, test data given stock data and sequence length
def load_data(stock, seq_len):
    data_raw = stock.to_numpy() # convert to numpy array
    data = []

    # create all possible sequences of length seq_len
    for index in range(len(data_raw) - seq_len):
        data.append(data_raw[index: index + seq_len])

    data = np.array(data);
    valid_set_size = int(np.round(valid_set_size_percentage/100*data.shape[0]));
    test_set_size = int(np.round(test_set_size_percentage/100*data.shape[0]));
    train_set_size = data.shape[0] - (valid_set_size + test_set_size);

    x_train = data[:train_set_size,:-1,:]
    y_train = data[:train_set_size,-1,:]
```

Figure 7(normalizing data)

4.2.6 Designing the neural network:

Once the dataset is uploaded, and after scaling down the data, the next step which is done is the creation of LSTM network. Keras is used to create the network and the TensorFlow is being utilized at the backend of Keras. Keras acts as a deep learning API which is written in the python language. It is executed on the top of machine learning on the TensorFlow platform. The primary reason to use the TensorFlow is to enable the fast experimentation. TensorFlow is a platform designed for the machine learning, it contains number of tools, libraries and various other resources which makes the process of LSTM creation a relatively an easier task or procedure.

For the purpose of designing a model for prediction, specific machine learning is used. For the sake of current research, Recurrent Neural Network is chosen primarily due to the nature of the data because the data of the stocks prices is in the form of time-series data. It has been studied those features of recurrent neural works in a better way than any other algorithm of machine learning to predict the time-series data. The design of the model is shown in the figure (1) which shows that it is a vital part of the model to first identify the train-test data then the scaling of the data is done using the Minmax Scalar for the purpose of training the data. Further a batch function is applied defined before the setup of recurrent neural networks. There are several parameters which needs to be set to obtain the potential results that includes the defining of placeholders, LSTM cell which is used to handle the short term and long-term memory, dynamic RNN and loss. These parameters help in calculation of Mean-squared error (MSE). The value of MSE helps to provide hint to the model that the prediction should be done, or the model should be re-run to tune the parameters of recurrent neural network. After the number of iterations, the model is finally capable of forecasting the stock price in a precise and accurate manner.

4.2.7 Training-testing of model:

After creation of network, the next step which is used includes the training of the network. The training of the data is done with the help of importing and scaling of the trained dataset. First the loss measure is obtained which is gathered in the form of root mean squared value. Then the training is done until the value is less than the 0.1 remains constant. The parameter tuning is used in this process of training until the best-trained network is obtained.

```
# parameters
n_steps = seq_len-1
n_inputs = 4
n_neurons = 200
n_outputs = 4
n_layers = 2
learning_rate = 0.001
batch_size = 50
n_epochs = 100
train_set_size = x_train.shape[0]
test_set_size = x_test.shape[0]
```

Figure 8(parameter tuning)

Data is always divided into two sets: one for the testing purpose and another one for training purpose. This separation is highly significant as the training process laid the foundation of the performance which is calculated on the basis of performance of the testing data.

```
# normalize stock
df_stock_norm = df_stock.copy()
df_stock_norm = normalize_data(df_stock_norm)

# create train, test data
seq_len = 20 # choose sequence length
x_train, y_train, x_valid, y_valid, x_test, y_test = load_data(df_stock_norm, seq_len)
print('x_train.shape = ',x_train.shape)
print('y_train.shape = ', y_train.shape)
print('x_valid.shape = ',x_valid.shape)
print('y_valid.shape = ', y_valid.shape)
print('x_test.shape = ', x_test.shape)
print('y_test.shape = ',y_test.shape)
```

Figure 9(training and testing)

The large bulk of data is usually used for training and the small amount is used for testing. Models are created on the basis of training dataset while the predictions are made on the basis of test datasets. After the training of the dataset, the next step which is used is the importing of the test data which is obtained as a result. The most important point while importing the test data is the values that is going to be scaled should be similar to the values obtained from the training of the data. This value should always be checked before moving on to import the test data. After importing of the trained data, trained model is applied to the test data and the resulting value is stored in the variable.

4.2.8 Visualization of the results:

Visualization is done after the process of testing and predicting values. The values obtained from the testing and the prediction are used in the scatter graph to visualize the results. There are two libraries which are used for visualization of the results that are obtained from the model. The first library used is the *mathplot library* which is used to visualize the results that are present in the console. Second library which is used is the *Plotly* which gives a clearer view of the results.

```
# use Basic RNN Cell
layers = [ tf.compat.v1.nn.rnn_cell.BasicRNNCell(num_units=n_neurons, activation=tf.nn.elu)
          for layer in range(n_layers)]

multi_layer_cell = tf.compat.v1.nn.rnn_cell.MultiRNNCell(layers)
rnn_outputs, states = tf.compat.v1.nn.dynamic_rnn(multi_layer_cell, X, dtype=tf.float32)

stacked_rnn_outputs = tf.reshape(rnn_outputs, [-1, n_neurons])
stacked_outputs = tf.compat.v1.layers.dense(stacked_rnn_outputs, n_outputs)
outputs = tf.reshape(stacked_outputs, [-1, n_steps, n_outputs])
outputs = outputs[:,n_steps-1,:] # keep only last output of sequence

loss = tf.reduce_mean(tf.square(outputs - y)) # loss function = mean squared error
optimizer = tf.compat.v1.train.AdamOptimizer(learning_rate=learning_rate)
training_op = optimizer.minimize(loss)

# run graph
with tf.compat.v1.Session() as sess:
    sess.run(tf.compat.v1.global_variables_initializer())
    for iteration in range(int(n_epochs*train_set_size/batch_size)):
        x_batch, y_batch = get_next_batch(batch_size) # fetch the next training batch
        sess.run(training_op, feed_dict={X: x_batch, y: y_batch})
        if iteration % int(5*train_set_size/batch_size) == 0:
            mse_train = loss.eval(feed_dict={X: x_train, y: y_train})
            mse_valid = loss.eval(feed_dict={X: x_valid, y: y_valid})
            print('%.2f epochs: MSE train/valid = %.6f/%.6f'%(
                iteration*batch_size/train_set_size, mse_train, mse_valid))

    y_train_pred = sess.run(outputs, feed_dict={X: x_train})
    y_valid_pred = sess.run(outputs, feed_dict={X: x_valid})
```

Figure 10(implementation of model)

```
0.00 epochs: MSE train/valid = 0.085160/0.004005
5.00 epochs: MSE train/valid = 0.000050/0.000017
10.00 epochs: MSE train/valid = 0.000080/0.000004
15.00 epochs: MSE train/valid = 0.000038/0.000002
20.00 epochs: MSE train/valid = 0.000035/0.000001
25.00 epochs: MSE train/valid = 0.000031/0.000003
30.00 epochs: MSE train/valid = 0.000023/0.000001
35.00 epochs: MSE train/valid = 0.000022/0.000001
40.00 epochs: MSE train/valid = 0.000033/0.000003
45.00 epochs: MSE train/valid = 0.000019/0.000001
50.00 epochs: MSE train/valid = 0.000033/0.000001
55.00 epochs: MSE train/valid = 0.000020/0.000002
60.00 epochs: MSE train/valid = 0.000021/0.000001
65.00 epochs: MSE train/valid = 0.000027/0.000002
70.00 epochs: MSE train/valid = 0.000032/0.000002
75.00 epochs: MSE train/valid = 0.000036/0.000002
80.00 epochs: MSE train/valid = 0.000029/0.000001
85.00 epochs: MSE train/valid = 0.000023/0.000004
90.00 epochs: MSE train/valid = 0.000024/0.000008
95.00 epochs: MSE train/valid = 0.000225/0.000025
```

Figure 11(Train and Valid Visualization)

```
## show predictions
plt.figure(figsize=(15, 5));
plt.subplot(1,2,1);

plt.plot(np.arange(y_train.shape[0]), y_train[:,ft], color='blue', label='train target')

plt.plot(np.arange(y_train.shape[0], y_train.shape[0]+y_valid.shape[0]), y_valid[:,ft],
         color='gray', label='valid target')

plt.plot(np.arange(y_train.shape[0]+y_valid.shape[0],
                   y_train.shape[0]+y_test.shape[0]+y_test.shape[0]),
         y_test[:,ft], color='black', label='test target')

plt.plot(np.arange(y_train_pred.shape[0]),y_train_pred[:,ft], color='red',
         label='train prediction')

plt.plot(np.arange(y_train_pred.shape[0], y_train_pred.shape[0]+y_valid_pred.shape[0]),
         y_valid_pred[:,ft], color='orange', label='valid prediction')

plt.plot(np.arange(y_train_pred.shape[0]+y_valid_pred.shape[0],
                   y_train_pred.shape[0]+y_valid_pred.shape[0]+y_test_pred.shape[0]),
         y_test_pred[:,ft], color='green', label='test prediction')
```

Figure 12(prediction)

4.2.9 Calculation of an efficiency of model:

After the successful completion of all steps, the last step which is left is the calculation of an efficiency of the model. The primary reason to calculate an efficiency is to find the reliability of the model which is trained and how the results are dependent on the model. The efficiency can be estimated in such a way if the system faces low loss, then it inferred the system to be more efficient. By the help of cost function, loss due to which the quality and the performance is reduced is evaluated. The key point while calculating the loss is the output value should be in the same format as the inputs for the purpose of comparison with the help of loss function. Further the optimization is performed using the Adam Optimizer. It is an optimization method which performs the gradient descent with the help of back propagation through the feature of time. In the project, optimizer. Minimize(loss) is used to present the trained dataset.

```
loss = tf.reduce_mean(tf.square(outputs - y)) # loss function = mean squared error
optimizer = tf.compat.v1.train.AdamOptimizer(learning_rate=learning_rate)
training_op = optimizer.minimize(loss)
```

Figure 13(LOSS FUNCTION)

The accuracy of the model can be calculated by the following formula shown in the figure (10).

```
from sklearn.metrics import mean_squared_error
accuracy=(1-mean_squared_error(y_test, y_test_pred))*100
print(f'Accuracy = {accuracy} % ')
```

FIGURE 14(ACCURACY FUNCTION)

4.3 Algorithm: Prediction using LSTM RNN

Input: BSE Sensex index historical prices
Output: prediction of Indian stock market and future price of the stocks.
Data <= BSE sensex index historical prices from kaggle.
Adj Close <= Adjacent Close Values Retrieved from Data.
Function preprocessing(Adj close,sequence_length)
Functional model

```
index_in_epoch = 0;
perm_array  = np.arange(x_train.shape[0])
np.random.shuffle(perm_array)
```

Function next batch

```
ef get_next_batch(batch_size):
   global index_in_epoch, x_train, perm_array
   start = index_in_epoch
   index_in_epoch += batch_size
```

Function Prediction // For plotting graphs.

```
plt.figure(figsize=(15, 5));
plt.subplot(1,2,1);

plt.plot(np.arange(y_train.shape[0]), y_train[:,ft], color='blue', label='train target')

plt.plot(np.arange(y_train.shape[0], y_train.shape[0]+y_valid.shape[0]), y_valid[:,ft],
        color='gray', label='valid target')

plt.plot(np.arange(y_train.shape[0]+y_valid.shape[0],
                   y_train.shape[0]+y_test.shape[0]+y_test.shape[0]),
        y_test[:,ft], color='black', label='test target')

plt.plot(np.arange(y_train_pred.shape[0]),y_train_pred[:,ft], color='red',
        label='train prediction')

plt.plot(np.arange(y_train_pred.shape[0], y_train_pred.shape[0]+y_valid_pred.shape[0]),
        y_valid_pred[:,ft], color='orange', label='valid prediction')

plt.plot(np.arange(y_train_pred.shape[0]+y_valid_pred.shape[0],
                   y_train_pred.shape[0]+y_valid_pred.shape[0]+y_test_pred.shape[0]),
        y_test_pred[:,ft], color='green', label='test prediction')

plt.title('past and future stock prices')
plt.xlabel('time [days]')
plt.ylabel('normalized price')
plt.legend(loc='best');

plt.subplot(1,2,2);

plt.plot(np.arange(y_train.shape[0], y_train.shape[0]+y_test.shape[0]),
        y_test[:,ft], color='black', label='test target')
```

```
plt.plot(np.arange(y_train_pred.shape[0], y_train_pred.shape[0]+y_test_pred.shape[0]),
         y_test_pred[:,ft], color='green', label='test prediction')

plt.title('future stock prices')
plt.xlabel('time [days]')
plt.ylabel('normalized price')
plt.legend(loc='best');
```

4.4 Import libraries:

There are various libraries which has been used in the project. The first library which is used in the project is the Keras library. This library is also used in the construction of LSTM Network. The most important thing which is imported is for the sequential model. This implies that the sequential model is used in the neural networks and not the Graph model. Another thing which is imported from the Keras library is the Dense layers. This inferred that dense layers is used in the neural networks. Then the most important thing which is used with the help of Keras Library is the LSTM. This further implies the use of LSTM in the neural network. This serves as the most important thing because the prediction relies on the LSTM network. The last thing which is imported from the Keras library is the Dropout. This is used to minimize the factor of overfiting in the NN layer thus providing an edge on the results.

```
import numpy as np
import pandas as pd
import math
import sklearn
import sklearn.preprocessing
import datetime
import os
import matplotlib.pyplot as plt
import tensorflow as tf
import tensorflow.compat.v1 as tf1


# split data in 80%/10%/10% train/validation/test sets
valid_set_size_percentage = 10
test_set_size_percentage = 10

#display parent directory and working directory
print(os.path.dirname(os.getcwd())+':', os.listdir(os.path.dirname(os.getcwd())));
print(os.getcwd()+':', os.listdir(os.getcwd()));
```

Figure 15(Library import)

4.5 Different graphs:

Output Screenshot:

```
<class 'pandas.core.frame.DataFrame'>
DatetimeIndex: 5396 entries, 2021-10-12 to 2000-01-03
Data columns (total 4 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   open    5396 non-null   float64
 1   close   5396 non-null   float64
 2   low     5396 non-null   float64
 3   high    5396 non-null   float64
dtypes: float64(4)
memory usage: 210.8 KB
```

Figure 16 (Showing Data type and length of CSV file)

```
<matplotlib.legend.Legend at 0x7fa69a573250>
```

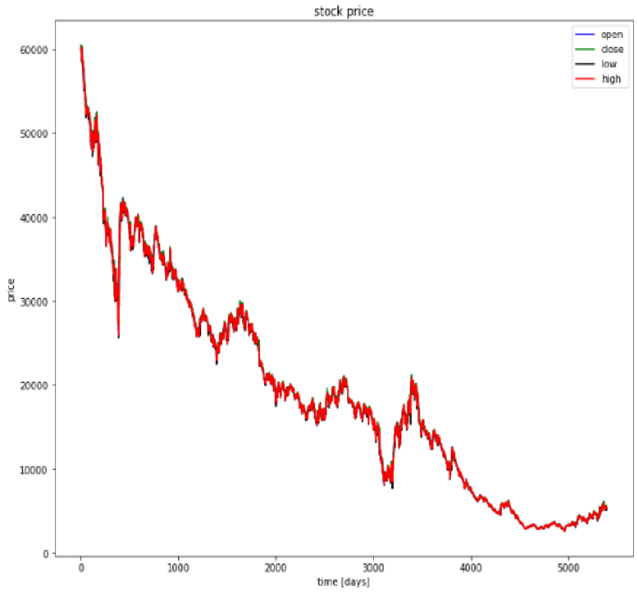


Figure 17(plotting columns of the dataset)

The Bove Figure 17 Contains the Graph with four Columns Open, Close, Low, High.

Open: Opening Margin of stock Price
Close: Closing Margin of Stock Price
Low: lowest Margin of Stock Price
High: Highest margin of Stock Price

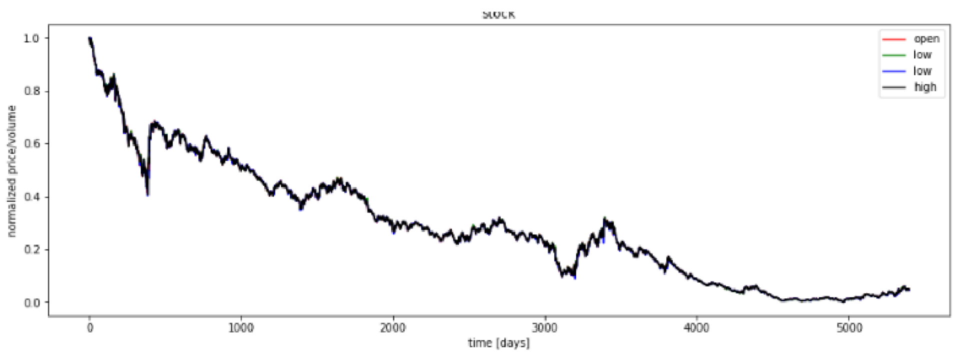


Figure 18 (Plotting the Volume of Stock with respect to Time/Days)

The Above Figure 18 Shows the Graphical View after the Normalizing the Price per Volume Per day with differentiate in four categories.

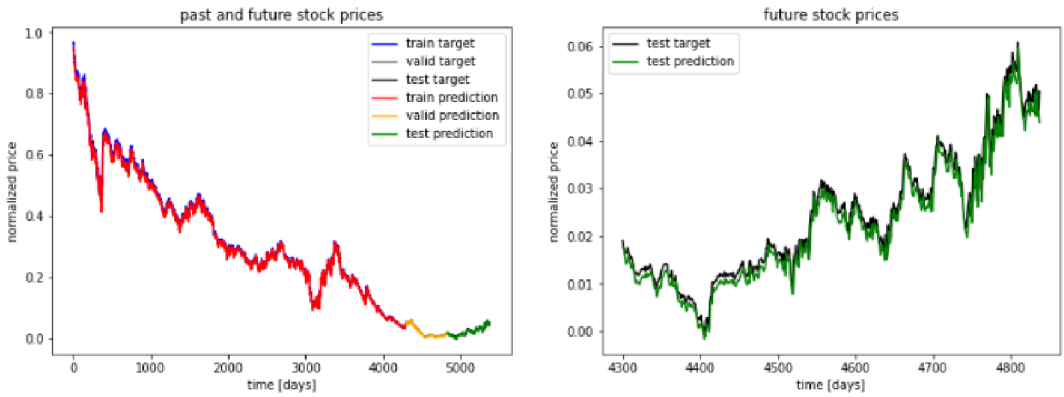


Figure 19(Plotting past and future of a Stock with respect to Time/Days)

The above Figures 19 Shows the Two Differential Graph after the Testing and training the Dataset Right Graphs Shows Future Stock Price and Past Stock price differencing Train, Valid, Test.
correct sign prediction for close - open price for train/valid/test: 0.48/0.30/0.36
Left Graph Shows the Future stock Price with test Target and test Prediction.

Prediction Error on the test file:

```
[227] prediction=linear_regressor.score(x_test, y_test)
      error_prediction=1-prediction
      print(error_prediction)

      0.0017547520149487905
```

```
[ ]  print(f'Accuracy = {accuracy} % ')

     Accuracy = 99.99960719990841 %
```

Figure 20 (Error Value & Accuracy)

The Above Figure 20 Shows Score for error is 0.001 which means that we achieved accuracy of 99.9 % in our forecasting.

To Find Out the Percentage error We use Below Formula and the Value we Predicted

$$PercentageError = \frac{(Actual - Forecast)}{Actual} \times 100$$

```
[ ]  from sklearn.metrics import mean_squared_error
     accuracy=(1-mean_squared_error(y_test, y_test_pred))*100
```

Figure 21(formula used in code)

The above figure 21 shows approximation error in a data value is the discrepancy between an exact value and some approximation to it. This error can be expressed as an absolute error (the numerical amount of the discrepancy) or as a relative error (the absolute error divided by the data value).

4.6 Comparison of Neural Networks for Stock Prediction:

There are three different types of neural networks. These are the artificial neural network (ANN), convolution neural network (CNN), and recurrent neural networks (RNN). Bing, Hao, and Zhang (2012) conducted a study that is associated with the stock market prediction by using artificial neural networks. Their study showed successful results as their trained artificial neural network proved to be very suitable for the prediction of the daily closing price, opening price, highest price, and lowest price of particularly the Shanghai Stock Exchange Composite Index. Furthermore, Guresen, Kayakutlu, and Daim (2011) also performed stock market prediction by using the trained models of artificial neural networks. These models include the multilayer perceptron (MLP), dynamic architecture for artificial neural networks (DAN2), GARCH MLP models, and the EGARCH MLP models. All these different models provided different results. Their results show that the MLP model gave more successful results than GARCH, EGARCH, and DAN2 models model showed only a 0.54% difference in the actual and the predicted stock price value of the NASDAQ stock exchange. Moreover, Qiu, Song, and Akagi (2016) also successfully predicted the stock market returns of the Japanese stock market by using artificial neural networks. Their experiment showed that the global search techniques, i.e., GA and SA, produced better accuracy than the traditional back propagation neural network algorithm.

Hoseinzade and Haratizadeh (2019) made use of two different convolution neural networks-based frameworks for stock market prediction. Their research study incorporates pooling and dropout methods for sampling and dataset training. In addition to this, Hoseinzade and Haratizadeh (2019) also used 3D CNNpred and 2D CNNpred to represent the input data to understand better. Thus, both the models entailed different stock market predictions but still produce more successful results than other baseline algorithms. The CNNpred frameworks outperformed the baseline algorithms such as F-measure and CNN-Cor within a range of 3 per cent to 11 per cent. Furthermore, Eapen, Bein, and Verma (2019) also used convolution neural networks for stock market prediction. CNN models that this study incorporates are support vector machine regressor models, single pipeline models, and the proposed multiple pipeline models. The results of this study show that multiple and single pipeline models produce better predictions than the support vector machine model. Among the multiple and single pipeline models multiple pipeline model performs approximately 9 per cent better and produced better results. Eapen, Bein, and Verma (2019) concluded that CNN models produce better stock market prediction than the traditional support vector machine model

Jahan (2018), in their research study, made use of the recurrent neural networks for stock market prediction. This study incorporates a sampling of the stock prices of five different companies. These are Amazon, Facebook, Microsoft, Google, and Netflix. The findings of this research revealed a strong and consistent link between actual and anticipated stock prices. Furthermore, the idea of the accuracy of this study can be obtained from the fact that it illustrated which is less than 1% percentage of error. Moreover, Pawar, Jalem and Tiwari (2019) performed stock market prediction using recurrent neural networks along with long short-term memory cells. They concluded in their study that RNN-LSTM produced better accuracy and prediction than the traditional machine learning algorithms. Along with these, Recurrent neural networks were also utilised by Li, Song, and Tao (2019) to forecast stock price movement. Their study proposed a multi-task RNN framework known as multi-task market price learner. This framework proved to be very effective and accurate while predicting three of the major Chinese stock market indexes.

Just like Li, Song, and Tao (2019), Pawar, Jalem and Tiwari (2019), and Jahan (2018), this research study also incorporates recurrent neural networks for predicting the stock market of India. In this dissertation also, RNN produced very favorable prediction results as the predicted values were approximately 80% similar to the actual values. Despite the proven success of convolution neural networks and artificial neural networks, the research chose recurrent neural networks for stock market prediction because, as Sharma et al. (2017) indicate, CNNs are better suited for processing images. Whereas recurrent neural networks make use of time-series information which associates that what happened last will affect what will happen next, which makes it suitable for processing temporal, sequential data, i.e., text and videos (Hewamalage, Bergmeir, and Bandara, 2017). CNN incorporates fixed-sized inputs and outputs (Rahmouni et al., 2017), and the same is the case with ANN (Walczak, 2019), whereas, in the case of RNN, the size of input and output can vary.

CHAPTER FIVE: RESULT & DISCUSSION

5.0 Introduction

This chapter of the dissertation study incorporates a thorough discussion of the final results. This study is associated with stock market prediction and efficiency analysis. The primary focus of this dissertation is towards predicting the stock market of particularly the housing finance industry of India. This chapter embodies a brief description of the dataset features and attributes that have been chosen by the researcher for proper stock market prediction. These are data, open, close, low and high.

5.1 Percentage of error

Prediction has become increasingly essential as technology has advanced. Therefore, forecasting is an essential component of the financial market. The stock market is one of the most extensively distributed sectors in the globe. Bonds, stocks, and various agreements between buyers and sellers are among the several functions contained in the stock. Because stock prices are dynamic, predicting them has become challenging. Various approaches have been employed to forecast stock prices and market trends. The following research has used neural networks to predict the stock market trend; while predicting, there is always uncertainty involved and not all predictions are accurate, hence percentage of error is always present when predicting future trends and forecasting events. The goal of every forecaster is to minimize the percentage of error to the maximum of its ability. A strong forecaster has the ability to determine the percentage error and considers the error when providing the results for its forecast.

A strategy known as "Percent Difference" or "Percentage Error" is one basic approach that many forecasters use to assess forecast accuracy. This is just the percentage difference between the actual volume and the anticipated volume. The following formula is used to determine the percentage of error in the model. The forecaster will forecast test data, and that test data will be tested with the actual data, through the formula, the overall percentage error for the model will be given. It is considered that less than 15% error is acceptable since the forecaster cannot be 100% accurate, so the margin of error is set to 15% and if the model has a percentage of error less than 15, then the forecaster is acceptable to be used in predicting stock market trends.

$$PercentageError = \frac{(Actual - Forecast)}{Actual} \times 100$$

Prediction and modelling are essential in data analysis. In today's society, the stock market is the most popular industry. The employment of traditional procedures may increase the likelihood of mathematical errors. The goal of prediction is to enhance the rate of investment and opportunity in the business. Predictions regarding the stock market can be produced through the use of algorithms. In the current project, the prediction is calculated to be 99.9% and the error is calculated to be less than 1%. In light of this, a model has been developed that uses the Recurrent neural network to anticipate the stock price but with a lower percentage of errors. Based on the results, efficiency is verified which greatly indicate the dependability of the model.

This dissertation study is associated with predicting the stock market and the efficiency analysis, particularly of India. Therefore, the dataset BSE Sensex Index historical price is used to evaluate the prediction error.

5.2 Discussion

The RNN method is based on learning from sequences, where the sequence is nothing more than a list of pairs (p_t,q_t), where p_t, resp. q_t denotes an input and the associated output, respectively, at a given time step t. Thus, we can have a consistent output value q_t=t throughout the whole series for different sorts of issues, or we can pick from a list of desired outcomes for each p_t. In addition, if the weight matrix values get too big, the gradient signal might become so strong that the learning scheme diverges. Exploding gradients is a term used to describe the latter. An intriguing technique for long-short term memory has been devised to address difficulties with lengthy sequences.

With two hidden layers, we created a recurrent neural network architecture based on LSTM. The uniqueness of this decision is that the first hidden layer has twice as many neurons as the second. This decision reflects a desire to enable the network to veil from the instances' unique information, finding generalized trends.

This dissertation study focuses on the Indian stock market, which has been very interesting in the last ten years. This study incorporates the exploitation of a ten-year dataset of the Indian stock exchange. Based on past data, our objective is to anticipate the direction of movement of the stock that we're interested in. In this dissertation, the stock prediction is based on a particular time window of open price, low price, close price, high price, and volume of traded shares.

Data visualization is associated with representing the data or information in the form of a chart, graph, or any other visual form. The significance of data visualization is that it makes it convenient to recognize the associated trends and patterns within the data. Thus, with the available large volume of data, data visualization not only provides a better and feasible understanding of data but also saves time to a great extent. Therefore, data scientists are constantly striving and engaging in developing machine learning algorithms for the better transformation of large data into correct and compelling visualizations. Some of the main methods in which data can be transformed into effective visualization are infographics, fever charts, Heatmap visualization, histogram, graph, frame diagrams, scatter plot, and area charts.

In this dissertation study, the researcher has mainly used a line graph to visualize the large dataset to predict the stock market for India. Abdulsalam, Adewole, and Jimoh (2011) predicted the stock prices of different banks by using line graphs for visualization. In addition, Umer, Awais, and Muzammul (2019) has also used a line graph for stock market prediction.

The prediction solution was built and tested on a hardware setup that included an NVIDIA Titan Xp 12 GB GPU, a 16-core Xeon E5-2609 CPU, and 32 GB RAM. In addition, the respective workstation was equipped with windows 10, python 3.9.2, Keras 2.4.0. Moreover, the system incorporates NumPy, dense, dropout, pandas, matplotlib, and MinMax Scaler for accurate stock prediction.

When the investor understands the prediction and the behavior of the market on the basis of rising and fall which occurs in the assets price. The next problem lies is the strategy of trading which tells that how much ratio they can share in the stocks. The investors should have prior knowledge of doing or making any type of investment in order to gain the benefit or profit. Therefore, the RNN model was found to be more efficient and beneficial at the same to give an estimate of what is the prediction rate in India by taking the dataset of BSE Sensex index. Moreover, there are several architectures that has been used in this regard which can be shown in the above figure. Since the greatest benefit of LSTM architecture is its best loss value so its behavior is also verified with the help of its inner architecture which contains the cells, layers and the activation function should be kept constant in the hidden layer. These varies from structure to structure which is used in RNN model.

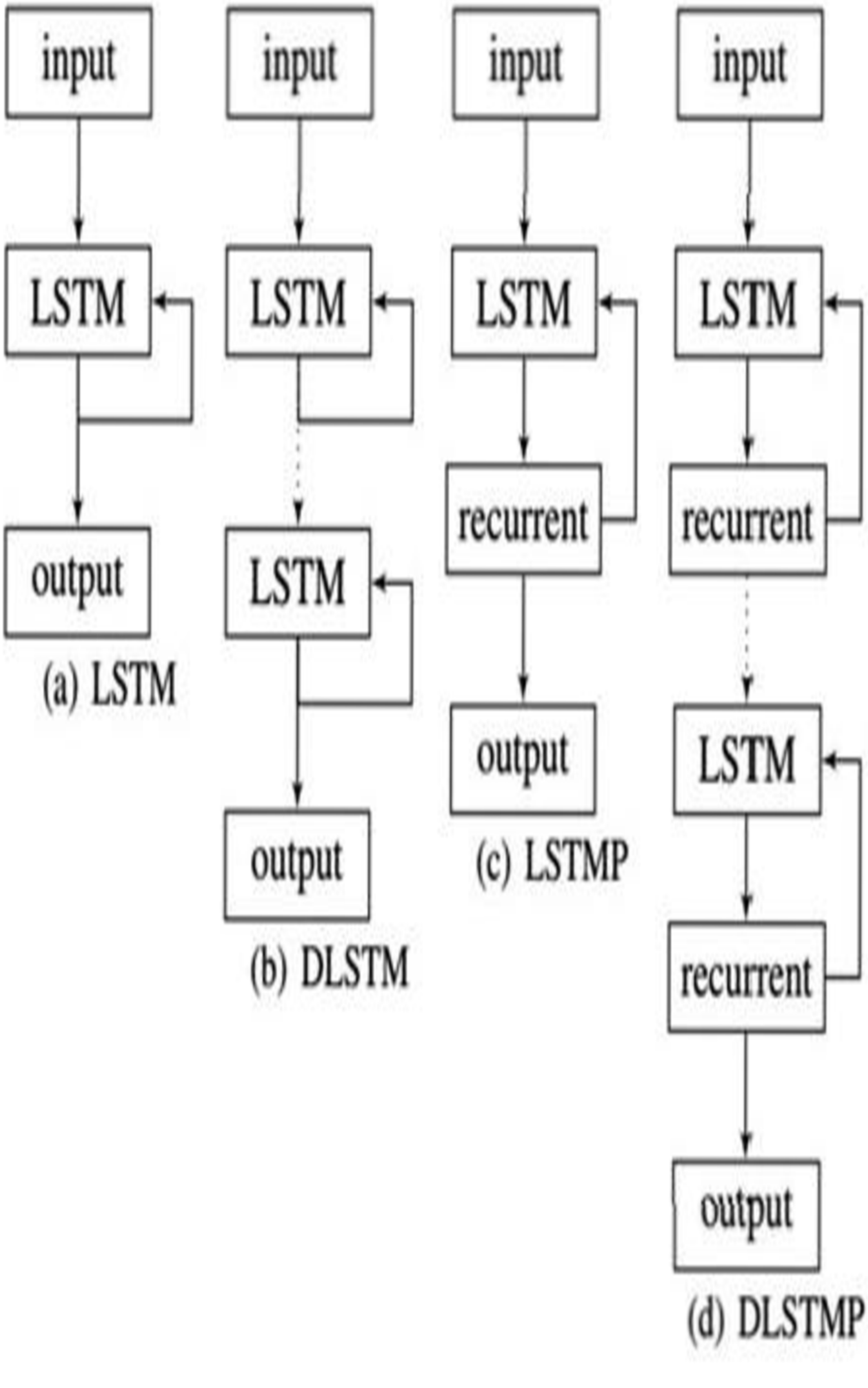


Figure 22(Various structures of LSTM)

CHAPTER 6 CONCLUSION AND RECOMMENDATIONS

6.1 Introduction

The chapter has focused on the concluding remarks regarding the analysis conducted towards the efficiency and prediction of the Stock markets and its eminent components involved by using a artificial intelligence methods using recurrent neural network platform using Python software. Machine learning techniques and their associated algorithm have been found very promising in assessing the stock markets recently. Thus, the study has focused on the key measures or prediction indicators found positive during the study. The chapter will highlight the significant implications, findings, and methods conducted with regard to predicting and assessing stock market efficiency. Based on the analysis, this chapter also discusses future goals, which can be effective in further growth. This chapter of the study is divided into significant subsections, including the summary regarding the essential elements found during the investigation, the recommendations related to the observed and highlighted scenarios, implications highlighted during the study, and lastly, the future directions for the research to analyze in-depth.

6.2 Summary of Findings

The primary aims and key objectives of this project involve the examination of the performance and feasibility of RNN in stock marketing forecasting. The study focused on the implementation of the RNN model for carrying out an in-depth analysis for predicting the stock markets based on neural channels (Shah et al., 2018). The results generated in the form of values were obtained from testing of various configurations, such as a multi-layered neural network is built by using a combination of mining data techniques. The data was analyzed based on major key steps, which included importing, visualizing, and training.

First, the neural network was trained on the stock using the Algorithm, which was used to predict closing price of market shares (Siregar and Wanto, 2017). The performance and its accuracy were then compared using various samples. For example, data that was gathered for Monday was matched with and tried to be predicted with second day's trend. This data was then saved in CSV for a simple retrieval as needed throughout the project. This information set was containing the daily trends of companies serving as the major information that can be used for future experiments.

Also, the findings obtained from the study through the values of analysts predicted that the efficiency of the system was directly proportional to the values of root mean square error. Therefore, this error has played a major role in predicting the stock price and market variations by minimizing errors up to a maximum limit.

Complex programming codes have been used to analyze the obtained data by establishing the predictions around it. This was achieved by using the built-in parameters in the algorithm responsible for decision-making. The algorithm was designed in such a way that any variation in the data results in shifting of the parameters accordingly with the data size and strength. The automatic shifting of variables was sounding to predict the current trends in the stock markets and is assumed as the best possible technique for quantitative analysis.

6.3 Recommendations

The use of an efficient and user-friendly algorithm can be used to identify the short-term cost of a single stock by increasing the rates of investment and opportunities related to business. The social algorithm can estimate the accurate rate of return and the business for making a sounding business environment towards the customers. Thus, it can be recommended that the updating in the data sets with respect to time and age would sound for the latest market holdings as the minimal changes in the sector can result in entirely various predictions which can be used to make a single decision for various enterprises working with the same motive. Also, the errors generated in outputs cannot be eradicated but can be minimized to their maximum limits. Thus, using root means square values can also effectively minimize the errors generated from the data, as it has been found promising based on various studies conducted. The use of root mean square error is found highly prevalent, and it could help make an excellent general purpose error metric for numerical predictions (Waqar et al., 2017).

6.4 Implications of Study

The study conducted regarding the use of efficient algorithms to assess the trends in stock markets has been found effective when methods of stock prediction were obtained through various literature sources. However, previous methods were promising in reducing the loss of error by about an average estimate of 20 per cent involving the use of neural networks (artificial) and convolutional networks (Li et al., 2018). But the study conducted is designed to reduce the error loss by about more average percentage as it is found promising that by reducing the loss error, the perception and efficiency of the stock markets can be analyzed with better results. Therefore, the study still lacks the prediction of the stock price with a lower percentage of error and needs more evidence with the use of a recurrent neural network for predicting the sounding results for future interventions. Further, the research has observed a limited set of data extracted for further processing through software, which is required in ample quantity to predict the stock market and its efficiency in the modern era. Thus, updating a project plan is still lacking, which needs to be updated with respect to the current variations in the stock markets in every region.

6.5 Future Research Directions

Based on the literature and evident texts found, there is a major growth of stock market trading currently in almost every region of the world. The popularity of stock market trading is rapidly increasing daily, which has been found effective in engaging the researchers to find the latest methods for predicting the use of new intervention or technology. Based on the analysis, the forecasting technique has been found quite beneficial and can also be used for future research in this domain. The technique associated with forecasting is not found to benefit the researchers only, but it has also been helping the investors and concerned persons associated with the stock market. Thus, for prediction against the indices of stock markets, this model of forecasting can be used with better and engaging accuracy in future research. The study, however, has used one of the most precision friendly forecasting technologies by using long and short-term memory unit and recurrent neural network, which would help analysts, stakeholders, or any concerned person showing interest in the investments related to stock markets by having a sounding knowledge of the situations concerning the future trends.

Thus, the forecasting model would be helpful to use in future for making decisions on investing in the markets based on the results generated from the model. Further, the model can be deployed to analyze the future market trends by making it more dynamic as possible by adding an additional attribute or feature. Also, the study can be used to predict investors' future sentiments on the topic of sensitive disclosures related to finances, such as the incorporation of changes in prices. Lastly, the study has focused on few recurrent neural network structures, but there are various other structures present that can be used in future to predict the stock market trends. Thus, future studies can be helpful in assessing them as well.

6.6 Conclusion

Based on the study's key findings, the research with respect to the assessment and decision making in prediction regarding the use of stock markets has been found very promising. There are also some significant implications reported that need to be further analyzed in continuing the approach. The trading associated with stock markets has been found convincing and positive in recent times as different tools discussed in the methodology section were found effective in determining the efficiency and predictors of the stock market. Using Kaggle to collect the Google dataset was convincing, based on relevant results generated. Thus, the study aims to use a primary dataset throughout the project due to convincing and satisfying results. The major recommendation regarding this project was to contain the daily change in ratios of stock price by storing it in a CSV file. The storing of data in this file would be helpful in making the import and export of data easier in Python. The stock sector is one of the widely spread sectors in the world. The different functions included in the stock are bonds, stocks and other agreements between buyers and sellers. With the generation of promising results from the findings and analysis, there are some limitations or implications observed during the study. As discussed in the above section, most of the datasets were not very current or were not up to the recent changes, which can be crucial for extracting the data and might predict inaccurate perceptions about the stock market. Also, the section has discussed the techniques for minimizing the errors or variations between the obtained output for further investigation. The minimization of the error would be helpful in the future for analyzing the data with sounding resolution and significant concentration.

References

Abdulsalam, S.O., Adewole, K.S. and Jimoh, R.G., 2011. Stock trend prediction using regression analysis–a data mining approach.

Abhishek, K., Singh, M.P., Ghosh, S. and Anand, A., 2012. Weather forecasting model using artificial neural network. Procedia Technology, 4, pp.311-318.

Aborisade, O.P., 2013. Data collection and new technology. *International Journal of Emerging Technologies in Learning (iJET)*, *8*(2), pp.48-52.

Ahangar, R.G., 2014. Stock Price Prediction of the Most Profitable Stock Exchange in the Asia during the Global Financial Crisis: A Comparative Study of Tehran Stock Exchange. *Trends in Applied Sciences Research*, *9*(2), p.61.

Alpaydin, E., 2014. Introduction to machine learning (adaptive computation and machine learning series).

Alves, L.G., Sigaki, H.Y., Perc, M. and Ribeiro, H.V., 2020. Collective dynamics of stock market efficiency. *Scientific reports*, *10*(1), pp.1-10.

Azzeh, M., Nassif, A.B. and Banitaan, S., 2018. Comparative analysis of soft computing techniques for predicting software effort-based use case points. *IET Software*, *12*(1), pp.19-29.

Bell, E., Bryman, A. and Harley, B., 2018. *Business research methods*. Oxford university press.

Billah, M., Waheed, S. and Hanifa, A., 2016, December. Stock market prediction using an improved training algorithm of neural network. In *2016 2nd International Conference on Electrical, Computer & Telecommunication Engineering (ICECTE)* (pp. 1-4). IEEE.

Bing, Y., Hao, J.K. and Zhang, S.C., 2012. Stock market prediction using artificial neural networks. In *Advanced Engineering Forum* (Vol. 6, pp. 1055-1060). Trans Tech Publications Ltd.

Bing, Y., Hao, J.K. and Zhang, S.C., 2012. Stock market prediction using artificial neural networks. In *Advanced Engineering Forum* (Vol. 6, pp. 1055-1060). Trans Tech Publications Ltd.

Bosco, J. and Khan, F., 2018. Stock Market Prediction and Efficiency Analysis using Recurrent Neural Network. Grin Verlag.

Bosco, J. and Khan, F., 2018. *Stock Market Prediction and Efficiency Analysis using Recurrent Neural Network*. Grin Verlag.

Bostanci, F., Kadioglu, E. and Sayilgan, G., 2018. Determinants of dividend payout decisions: a dynamic panel data analysis of Turkish stock market. *International journal of financial* *studies*, *6*(4), p.93.

Bougie, R. and Sekaran, U., 2016. *Research Methods for Business: A Skill Building 7 ed*. Wily.
Jeong, E., Park, N., Choi, Y., Park, R.W. and Yoon, D., 2018. Machine learning model combining features from algorithms with different analytical methodologies to detect laboratory-event-related adverse drug reaction signals. *PloS one*, *13*(11), p.e0207749.

Brickley, D., Burgess, M. and Noy, N., 2019, May. Google Dataset Search: Building a search engine for datasets in an open Web ecosystem. In The World Wide Web Conference (pp. 1365-1375).

Bukhari, A.H., Raja, M.A.Z., Sulaiman, M., Islam, S., Shoaib, M. and Kumam, P., 2020. Fractional neuro-sequential ARFIMA-LSTM for financial market forecasting. IEEE Access, 8, pp.71326-71338.

Cartea, Á. and Jaimungal, S., 2016. A closed-form execution strategy to target volume weighted average price. *SIAM Journal on Financial Mathematics*, *7*(1), pp.760-785.

Chang, C.Y., 2011. The relationship between the 52-week high of an individual stock and stock market index level: Evidence from Taiwan. *Journal of International Financial Markets, Institutions and Money*, *21*(1), pp.14-27.

Chawla, A., Lee, B., Fallon, S. and Jacob, P., 2018, September. Host based intrusion detection system with combined CNN/RNN model. In Joint European Conference on Machine Learning and Knowledge Discovery in Databases (pp. 149-158). Springer, Cham.

Chou, J.S. and Nguyen, T.K., 2018. Forward forecast of stock price using sliding-window metaheuristic-optimized machine-learning regression. *IEEE Transactions on Industrial Informatics*, *14*(7), pp.3132-3142.

Eapen, J., Bein, D. and Verma, A., 2019, January. Novel deep learning model with CNN and bi-directional LSTM for improved stock market index prediction. In *2019 IEEE 9th annual computing and communication workshop and conference (CCWC)* (pp. 0264-0270). IEEE.

Fathi, O., 2019. Time series forecasting using a hybrid ARIMA and LSTM model. *Velvet Consulting*.

Gao, Q., 2016. Stock market forecasting using recurrent neural network (Doctoral dissertation, University of Missouri--Columbia).

Geoffrey, M., 2019. Essential of Research design and methodology.

Guresen, E., Kayakutlu, G. and Daim, T.U., 2011. Using artificial neural network models in stock market index prediction. *Expert Systems with Applications*, *38*(8), pp.10389-10397.

Heaton, J.B., 2021. Bankruptcy Prediction Using Only the Face Value of Debt and Stock Market Capitalization in a Single Ratio. *Available at SSRN 3773332*.

Hegazy, O., Soliman, O.S. and Salam, M.A., 2014. A machine learning model for stock market prediction. *arXiv preprint arXiv:1402.7351*.

Hewamalage, H., Bergmeir, C. and Bandara, K., 2021. Recurrent neural networks for time series forecasting: Current status and future directions. *International Journal of Forecasting*, *37*(1), pp.388-427.

Hiransha, M., Gopalakrishnan, E.A., Menon, V.K. and Soman, K.P., 2018. NSE stock market prediction using deep-learning models. Procedia computer science, 132, pp.1351-1362.
Hiransha, M., Gopalakrishnan, E.A., Menon, V.K. and Soman, K.P., 2018. NSE stock market prediction using deep-learning models. *Procedia computer science*, *132*, pp.1351-1362.
Hoseinzade, E. and Haratizadeh, S., 2019. CNNpred: CNN-based stock market prediction using a diverse set of variables. *Expert Systems with Applications*, *129*, pp.273-285.

Hu, Z., Zhao, Y. and Khushi, M., 2021. A survey of forex and stock price prediction using deep learning. *Applied System Innovation*, *4*(1), p.9.

Jahan, I., 2018. Stock price prediction using recurrent neural networks.
Jahan, I., 2018. Stock price prediction using recurrent neural networks.
Khalifa, N.E.M., Smarandache, F., Manogaran, G. and Loey, M., 2021. A study of the neutrosophic set significance on deep transfer learning models: An experimental case on a limited covid-19 chest x-ray dataset. Cognitive Computation, pp.1-10.

Kim, K.J. and Lee, W.B., 2004. Stock market prediction using artificial neural networks with optimal feature transformation. *Neural computing & applications*, *13*(3), pp.255-260.

Kumar Deva, B., Sharon, S. and Maria, J.E., 2015. Empirical Study on Effects of the Lok Sabha Election on Stock Market Performance (BSE SENSEX). *Research Journal of Management Sciences* ____*ISSN*, *2319*, p.1171.

Kumar, J., Goomer, R. and Singh, A.K., 2018. Long short-term memory recurrent neural network (LSTM-RNN) based workload forecasting model for cloud datacenters. Procedia Computer Science, 125, pp.676-682.

Li, C., Song, D. and Tao, D., 2019, July. Multi-task recurrent neural networks and higher-order Markov random fields for stock price movement prediction: Multi-task RNN and higer-order

MRFs for stock price classification. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining* (pp. 1141-1151).

Li, X., Ding, Q. and Sun, J.Q., 2018. Remaining useful life estimation in prognostics using deep convolution neural networks. Reliability Engineering & System Safety, 172, pp.1-11.

Liu, Y., Qin, Z., Li, P. and Wan, T., 2017, June. Stock volatility prediction using recurrent neural networks with sentiment analysis. In *International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems* (pp. 192-201). Springer, Cham.

Maknickienė, N., Rutkauskas, A.V. and Maknickas, A., 2011. Investigation of financial market prediction by recurrent neural network. Innovative Technologies for Science, Business and Education, 2(11), pp.3-8.

Malhotra, R., 2015. A systematic review of machine learning techniques for software fault prediction. Applied Soft Computing, 27, pp.504-518.

Marczyk, G.R., DeMatteo, D. and Festinger, D., 2021. *Essentials of research design and methodology*. John Wiley & Sons, Inc.

McDermott, P.L. and Wikle, C.K., 2019. Bayesian recurrent neural network models for forecasting and quantifying uncertainty in spatial-temporal data. Entropy, 21(2), p.184.

McDermott, P.L. and Wikle, C.K., 2019. Bayesian recurrent neural network models for forecasting and quantifying uncertainty in spatial-temporal data. *Entropy*, *21*(2), p.184.

Medsker, L.R. and Jain, L.C., 2001. Recurrent neural networks. Design and Applications, 5, pp.64-67.

Nabipour, M., Nayyeri, P., Jabani, H., Mosavi, A. and Salwana, E., 2020. Deep learning for stock market prediction. *Entropy*, *22*(8), p.840.

Naik, N. and Mohan, B.R., 2019, May. Stock price movements classification using machine and deep learning techniques-the case study of indian stock market. In *International Conference on Engineering Applications of Neural Networks* (pp. 445-452). Springer, Cham.

Nassif, A.B., Azzeh, M., Capretz, L.F. and Ho, D., 2016. Neural network models for software development effort estimation: a comparative study. *Neural Computing and Applications*, *27*(8), pp.2369-2381.

Nti, I.K., Adekoya, A.F. and Weyori, B.A., 2020. A systematic review of fundamental and technical analysis of stock market predictions. *Artificial Intelligence Review*, *53*(4), pp.3007-3057.

Obthong, M., Tantisantiwong, N., Jeamwatthanachai, W. and Wills, G., 2020. A survey on machine learning for stock price prediction: algorithms and techniques.

Oncharoen, P. and Vateekul, P., 2018, August. Deep learning for stock market prediction using event embedding and technical indicators. In *2018 5th International Conference on Advanced Informatics: Concept Theory and Applications (ICAICTA)* (pp. 19-24). IEEE.

Osamwonyi, I.O. and Kasimu, A., 2013. Stock market and economic growth in Ghana, Kenya and Nigeria. *International Journal of Financial Research*, *4*(2), p.83.

Paradis, E., O'Brien, B., Nimmon, L., Bandiera, G. and Martimianakis, M.A., 2016. Design: Selection of data collection methods. *Journal of graduate medical education*, *8*(2), pp.263-264.

Patel, S.D., Quadros, D., Patil, V. and Saxena, H., 2017. Stock prediction using neural networks. International Journal of Engineering and Management Research (IJEMR), 7(2), pp.490-493.

Pawar, K., Jalem, R.S. and Tiwari, V., 2019. Stock market price prediction using LSTM RNN. In *Emerging Trends in Expert Applications and Security* (pp. 493-503). Springer, Singapore.

Pejić Bach, M., Krstić, Ž., Seljan, S. and Turulja, L., 2019. Text mining for big data analysis in financial sector: A literature review. *Sustainability*, *11*(5), p.1277.

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

Qiu, M., Song, Y. and Akagi, F., 2016. Application of artificial neural network for the prediction of stock market returns: The case of the Japanese stock market. *Chaos, Solitons & Fractals*, *85*, pp.1-7.

Radityo, A., Munajat, Q. and Budi, I., 2017, October. Prediction of Bitcoin exchange rate to American dollar using artificial neural network methods. In *2017 International Conference on Advanced Computer Science and Information Systems (ICACSIS)* (pp. 433-438). IEEE.

Rahmouni, N., Nozick, V., Yamagishi, J. and Echizen, I., 2017, December. Distinguishing computer graphics from natural images using convolution neural networks. In *2017 IEEE Workshop on Information Forensics and Security (WIFS)* (pp. 1-6). IEEE.

Raj, J.S. and Ananthi, J.V., 2019. Recurrent neural networks and nonlinear prediction in support vector machines. *Journal of Soft Computing Paradigm (JSCP)*, *1*(01), pp.33-40.

Roondiwala, M., Patel, H. and Varma, S., 2017. Predicting stock prices using LSTM. International Journal of Science and Research (IJSR), 6(4), pp.1754-1756.

Selvamuthu, D., Kumar, V. and Mishra, A., 2019. Indian stock market prediction using artificial neural networks on tick data. *Financial Innovation*, *5*(1), pp.1-12.

Selvin, S., Vinayakumar, R., Gopalakrishnan, E.A., Menon, V.K. and Soman, K.P., 2017, September. Stock price prediction using LSTM, RNN and CNN-sliding window model. In *2017* *international conference on advances in computing, communications and informatics (icacci)* (pp. 1643-1647). IEEE.

Shadbolt, J. and Taylor, J.G., 2002. Input selection. In *Neural Networks and the Financial Markets* (pp. 77-83). Springer, London.

Shah, D., Campbell, W. and Zulkernine, F.H., 2018, December. A comparative study of LSTM and DNN for stock market forecasting. In 2018 IEEE International Conference on Big Data (Big Data) (pp. 4148-4155). IEEE.

Sharma, A., Bhuriya, D. and Singh, U., 2017, April. Survey of stock market prediction using machine learning approach. In *2017 International conference of electronics, communication, and aerospace technology (ICECA)* (Vol. 2, pp. 506-509). IEEE.

Sharma, J., Granmo, O.C., Goodwin, M. and Fidje, J.T., 2017, August. Deep convolutional neural networks for fire detection in images. In *International conference on engineering applications of neural networks* (pp. 183-193). Springer, Cham.

Sileyew, K.J., 2019. Research design and methodology. In *Cyberspace*. IntechOpen.

Sim, H.S., Kim, H.I. and Ahn, J.J., 2019. Is deep learning for image recognition applicable to stock market prediction? *Complexity*, *2019*.

Singh, A., 2018. Stock Prices Prediction Using Machine Learning and Deep Learning Techniques (with Python codes). *OCTOBER 25, 2018*.

Siregar, S.P. and Wanto, A., 2017. Analysis of artificial neural network accuracy using backpropagation algorithm in predicting process (forecasting). IJISTECH (International Journal of Information System & Technology), 1(1), pp.34-42.

Statistical hypothesis testing is associated with utilizing observed data for formulating conclusions for a particular claim that is associated with a larger population. A statistical hypothesis is a testable hypothesis. It is testable based on the observation of a procedure that has been modelled using a combination of various random variables. The statistical hypothesis test is one of the various methods of statistical inference. This method primarily incorporates the comparison of two different statistical datasets or two such datasets, one is derived from sampling, and the other is a synthetic dataset obtained from an ideal model. Further, the process entails proposing a hypothesis regarding the statistical relationship between the previously compared datasets. This proposed hypothesis is then compared with an ideal hypothesis that is associated with proposing no relationship between the identified datasets. If the relationship between the datasets follows the threshold probability known as the significance level, only then the relationship is considered statistically significant. Thus, the significance of statistical hypothesis tests is that it aids in determining the respective end results of a study that would lead to declining the null hypothesis in terms of a pre-established level of significance. The identification of conceptual errors associated with misunderstanding the context is one of the ways of differentiating a null hypothesis from an alternative hypothesis. These conceptual errors are usually categorized as type 1 and type 2.

Stošić, D., Stošić, D., Stošić, T. and Stanley, H.E., 2015. Multifractal properties of price change and volume change of stock market indices. *Physica A: Statistical Mechanics and its Applications*, *428*, pp.46-51.

Sun, W., Venkatraman, A., Gordon, G.J., Boots, B. and Bagnell, J.A., 2017, July. Deeply aggrevated: Differentiable imitation learning for sequential prediction. In International Conference on Machine Learning (pp. 3309-3318). PMLR.

Tirea, M., Tandau, I. and Negru, V., 2011, September. Multi-agent stock trading algorithm model. In *2011 13th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing* (pp. 365-372). IEEE.

Tsai, C.F. and Wang, S.P., 2009, March. Stock price forecasting by hybrid machine learning techniques. In Proceedings of the international multiconference of engineers and computer scientists (Vol. 1, No. 755, p. 60).

Umer, M., Awais, M. and Muzammul, M., 2019. Stock Market Prediction Using Machine Learning (ML) Algorithms. *ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal*, *8*(4), pp.97-116.

Vadlamudi, S., 2017. Stock Market Prediction using Machine Learning: A Systematic Literature Review. *American Journal of Trade and Policy*, *4*(3), pp.123-128.

Walczak, S., 2019. Artificial neural networks. In *Advanced Methodologies and Technologies in Artificial Intelligence, Computer Simulation, and Human-Computer Interaction* (pp. 40-53). IGI Global.

Waqar, M., Dawood, H., Guo, P., Shahnawaz, M.B. and Ghazanfar, M.A., 2017, December. Prediction of stock market by principal component analysis. In 2017 13th International Conference on Computational Intelligence and Security (CIS) (pp. 599-602). IEEE.

Yetis, Y., Kaplan, H. and Jamshidi, M., 2014, August. Stock market prediction by using artificial neural network. In *2014 World Automation Congress (WAC)* (pp. 718-722). IEEE.

Yoo, P.D., Kim, M.H. and Jan, T., 2005, November. Machine learning techniques and use of event information for stock market prediction: A survey and evaluation. In *International Conference on Computational Intelligence for Modelling, Control and Automation and*

International Conference on Intelligent Agents, Web Technologies, and Internet Commerce (CIMCA-IAWTIC'06) (Vol. 2, pp. 835-841). IEEE.

Zhang, G.P., 2003. Time series forecasting using a hybrid ARIMA and neural network model. *Neurocomputing*, *50*, pp.159-175.

Zhang, Z., Shen, Y., Zhang, G., Song, Y. and Zhu, Y., 2017, November. Short-term prediction for opening price of stock market based on self-adapting variant PSO-Elman neural network. In *2017 8th IEEE International Conference on Software Engineering and Service Science (ICSESS)* (pp. 225-228). IEEE.

Stefan Lauren Dra. Harlili S., M.Sc "Stock Trend Prediction Using Simple Moving Average Supported by News Classification" 2014 International Conference of Advanced Informatics: Concept, Theory, and Application (ICAICTA).

Appendix:

Code:

```
import numpy as np
import pandas as pd
import math
import sklearn
import sklearn.preprocessing
import datetime
import os
import matplotlib.pyplot as plt
import tensorflow as tf
import tensorflow.compat.v1 as tf1


# split data in 80%/10%/10% train/validation/test sets
valid_set_size_percentage = 10
test_set_size_percentage = 10

#display parent directory and working directory
print(os.path.dirname(os.getcwd())+':', os.listdir(os.path.dirname(os.getcwd())));
print(os.getcwd()+':', os.listdir(os.getcwd()));

# import all stock prices
df = pd.read_excel("/content/BSE Sensex Index Historical Prices.xlsx", index_col = 0)
df.info()
df.head()

# number of different stocks
# print('\nnumber of different stocks: ', len(list(set(df.symbol))))
# print(list(set(df.symbol))[:10])

plt.figure(figsize=(25, 10));
plt.subplot(1,2,1);
df.columns = ['open', 'high', 'low', 'close']
plt.plot(df['open'].values, color='blue', label='open')
plt.plot(df['close'].values, color='green', label='close')
plt.plot(df['low'].values, color='black', label='low')
plt.plot(df['high'].values, color='red', label='high')
plt.title('stock price')
plt.xlabel('time [days]')
plt.ylabel('price')
plt.legend(loc='best')


def normalize_data(df):
  min_max_scaler = sklearn.preprocessing.MinMaxScaler()
  df['open'] = min_max_scaler.fit_transform(df.open.values.reshape(-1,1))
```

```
    df['high'] = min_max_scaler.fit_transform(df.high.values.reshape(-1,1))
    df['low'] = min_max_scaler.fit_transform(df.low.values.reshape(-1,1))
    df['close'] = min_max_scaler.fit_transform(df['close'].values.reshape(-1,1))
    return df

# function to create train, validation, test data given stock data and sequence length
def load_data(stock, seq_len):
    data_raw = stock.to_numpy() # convert to numpy array
    data = []

    # create all possible sequences of length seq_len
    for index in range(len(data_raw) - seq_len):
        data.append(data_raw[index: index + seq_len])

    data = np.array(data);
    valid_set_size = int(np.round(valid_set_size_percentage/100*data.shape[0]));
    test_set_size = int(np.round(test_set_size_percentage/100*data.shape[0]));
    train_set_size = data.shape[0] - (valid_set_size + test_set_size);

    x_train = data[:train_set_size,:-1,:]
    y_train = data[:train_set_size,-1,:]

    x_valid = data[train_set_size:train_set_size+valid_set_size,:-1,:]
    y_valid = data[train_set_size:train_set_size+valid_set_size,-1,:]

    x_test = data[train_set_size+valid_set_size:,:-1,:]
    y_test = data[train_set_size+valid_set_size:,-1,:]

    return [x_train, y_train, x_valid, y_valid, x_test, y_test]

# choose one stock
df_stock = df.copy()
# df_stock.drop(['symbol'],1,inplace=True)
# df_stock.drop(['volume'],1,inplace=True)

cols = list(df_stock.columns.values)
print('df_stock.columns.values = ', cols)

# normalize stock
df_stock_norm = df_stock.copy()
df_stock_norm = normalize_data(df_stock_norm)

# create train, test data
seq_len = 20 # choose sequence length
x_train, y_train, x_valid, y_valid, x_test, y_test = load_data(df_stock_norm, seq_len)
print('x_train.shape = ',x_train.shape)
print('y_train.shape = ', y_train.shape)
```

```
print('x_valid.shape = ',x_valid.shape)
print('y_valid.shape = ', y_valid.shape)
print('x_test.shape = ', x_test.shape)
print('y_test.shape = ',y_test.shape)

plt.figure(figsize=(15, 5));
plt.plot(df_stock_norm.open.values, color='red', label='open')
plt.plot(df_stock_norm.close.values, color='green', label='low')
plt.plot(df_stock_norm.low.values, color='blue', label='low')
plt.plot(df_stock_norm.high.values, color='black', label='high')
#plt.plot(df_stock_norm.volume.values, color='gray', label='volume')
plt.title('stock')
plt.xlabel('time [days]')
plt.ylabel('normalized price/volume')
plt.legend(loc='best')
plt.show()

index_in_epoch = 0;
perm_array  = np.arange(x_train.shape[0])
np.random.shuffle(perm_array)

# function to get the next batch
def get_next_batch(batch_size):
    global index_in_epoch, x_train, perm_array
    start = index_in_epoch
    index_in_epoch += batch_size

    if index_in_epoch > x_train.shape[0]:
        np.random.shuffle(perm_array) # shuffle permutation array
        start = 0 # start next epoch
        index_in_epoch = batch_size

    end = index_in_epoch
    return x_train[perm_array[start:end]], y_train[perm_array[start:end]]

# parameters
n_steps = seq_len-1
n_inputs = 4
n_neurons = 200
n_outputs = 4
n_layers = 2
learning_rate = 0.001
batch_size = 50
n_epochs = 100
train_set_size = x_train.shape[0]
test_set_size = x_test.shape[0]
```

```
tf.compat.v1.reset_default_graph()
tf.compat.v1.disable_eager_execution()
X = tf1.placeholder(tf1.float32, [None, n_steps, n_inputs])
y = tf1.placeholder(tf1.float32, [None, n_outputs])

# use Basic RNN Cell

layers = [ tf.compat.v1.nn.rnn_cell.BasicRNNCell(num_units=n_neurons, activation=tf.nn.elu)
        for layer in range(n_layers)]

# use Basic LSTM Cell
#layers = [tf.contrib.rnn.BasicLSTMCell(num_units=n_neurons, activation=tf.nn.elu)
#         for layer in range(n_layers)]

# use LSTM Cell with peephole connections
#layers = [tf.contrib.rnn.LSTMCell(num_units=n_neurons,
#                              activation=tf.nn.leaky_relu, use_peepholes = True)
#         for layer in range(n_layers)]

# use GRU cell
#layers = [tf.contrib.rnn.GRUCell(num_units=n_neurons, activation=tf.nn.leaky_relu)
#         for layer in range(n_layers)]

multi_layer_cell = tf.compat.v1.nn.rnn_cell.MultiRNNCell(layers)
rnn_outputs, states = tf.compat.v1.nn.dynamic_rnn(multi_layer_cell, X, dtype=tf.float32)

stacked_rnn_outputs = tf.reshape(rnn_outputs, [-1, n_neurons])
stacked_outputs = tf.compat.v1.layers.dense(stacked_rnn_outputs, n_outputs)
outputs = tf.reshape(stacked_outputs, [-1, n_steps, n_outputs])
outputs = outputs[:,n_steps-1,:] # keep only last output of sequence

loss = tf.reduce_mean(tf.square(outputs - y)) # loss function = mean squared error
optimizer = tf.compat.v1.train.AdamOptimizer(learning_rate=learning_rate)
training_op = optimizer.minimize(loss)
```

```
# run graph
with tf.compat.v1.Session() as sess:
    sess.run(tf.compat.v1.global_variables_initializer())
    for iteration in range(int(n_epochs*train_set_size/batch_size)):
        x_batch, y_batch = get_next_batch(batch_size) # fetch the next training batch
        sess.run(training_op, feed_dict={X: x_batch, y: y_batch})
        if iteration % int(5*train_set_size/batch_size) == 0:
            mse_train = loss.eval(feed_dict={X: x_train, y: y_train})
            mse_valid = loss.eval(feed_dict={X: x_valid, y: y_valid})
            print('%.2f epochs: MSE train/valid = %.6f/%.6f'%(
                iteration*batch_size/train_set_size, mse_train, mse_valid))

    y_train_pred = sess.run(outputs, feed_dict={X: x_train})
    y_valid_pred = sess.run(outputs, feed_dict={X: x_valid})
    y_test_pred = sess.run(outputs, feed_dict={X: x_test})

y_train.shape

ft = 0 # 0 = open, 1 = close, 2 = highest, 3 = lowest

## show predictions
plt.figure(figsize=(15, 5));
plt.subplot(1,2,1);

plt.plot(np.arange(y_train.shape[0]), y_train[:,ft], color='blue', label='train target')

plt.plot(np.arange(y_train.shape[0], y_train.shape[0]+y_valid.shape[0]), y_valid[:,ft],
         color='gray', label='valid target')

plt.plot(np.arange(y_train.shape[0]+y_valid.shape[0],
                   y_train.shape[0]+y_test.shape[0]+y_test.shape[0]),
         y_test[:,ft], color='black', label='test target')

plt.plot(np.arange(y_train_pred.shape[0]),y_train_pred[:,ft], color='red',
         label='train prediction')
```

```
plt.plot(np.arange(y_train_pred.shape[0], y_train_pred.shape[0]+y_valid_pred.shape[0]),
        y_valid_pred[:,ft], color='orange', label='valid prediction')

plt.plot(np.arange(y_train_pred.shape[0]+y_valid_pred.shape[0],
                y_train_pred.shape[0]+y_valid_pred.shape[0]+y_test_pred.shape[0]),
        y_test_pred[:,ft], color='green', label='test prediction')

plt.title('past and future stock prices')
plt.xlabel('time [days]')
plt.ylabel('normalized price')
plt.legend(loc='best');

plt.subplot(1,2,2);

plt.plot(np.arange(y_train.shape[0], y_train.shape[0]+y_test.shape[0]),
        y_test[:,ft], color='black', label='test target')

plt.plot(np.arange(y_train_pred.shape[0], y_train_pred.shape[0]+y_test_pred.shape[0]),
        y_test_pred[:,ft], color='green', label='test prediction')

plt.title('future stock prices')
plt.xlabel('time [days]')
plt.ylabel('normalized price')
plt.legend(loc='best');

corr_price_development_train = np.sum(np.equal(np.sign(y_train[:,1]-y_train[:,0]),
          np.sign(y_train_pred[:,1]-y_train_pred[:,0])).astype(int)) / y_train.shape[0]
corr_price_development_valid = np.sum(np.equal(np.sign(y_valid[:,1]-y_valid[:,0]),
          np.sign(y_valid_pred[:,1]-y_valid_pred[:,0])).astype(int)) / y_valid.shape[0]
corr_price_development_test = np.sum(np.equal(np.sign(y_test[:,1]-y_test[:,0]),
          np.sign(y_test_pred[:,1]-y_test_pred[:,0])).astype(int)) / y_test.shape[0]

print('correct sign prediction for close - open price for train/valid/test: %.2f/%.2f/%.2f'%(
   corr_price_development_train, corr_price_development_valid, corr_price_development_test)
   )
from sklearn.metrics import mean_squared_error
accuracy=(1-mean_squared_error(y_test, y_test_pred))*100
print(f'Accuracy = {accuracy} % ')
```