**ABSTRACT**

The project "Stock Price Prediction Using LSTM" aims to develop a predictive model for forecasting stock prices by leveraging Long Short-Term Memory (LSTM) neural networks, a type of recurrent neural network (RNN) known for its ability to capture sequential dependencies in data. Stock price prediction is a crucial task in financial markets, as accurate forecasts can inform investment decisions, risk management, and portfolio optimization.

In this project, historical stock price and related financial data are collected and pre-processed. The LSTM architecture is employed to learn and analyze patterns and trends in the data. LSTM networks are well-suited for this task because they can capture both short-term and long-term dependencies, which are essential in financial time series analysis.

The process involves data normalization, feature engineering, and the creation of a suitable training dataset for the LSTM model. The model is trained using the historical data and tested on a separate dataset to evaluate its predictive performance. Various evaluation metrics, such as Mean Absolute Error (MAE) and Root Mean Square Error (RMSE), are used to assess the model's accuracy. The results of the project aim to provide investors, traders, and financial analysts with a tool for making more informed decisions in the stock market. Additionally, it explores the potential of deep learning and recurrent neural networks in financial forecasting, highlighting their ability to capture complex temporal patterns in stock price data.

Ultimately, this project contributes to the growing field of financial technology (FinTech) and machine learning in finance, offering a valuable tool for individuals and institutions seeking to enhance their stock market predictions and, in turn, optimize their investment strategies.

**TABLE OF CONTENTS**

**1. INTRODUCTION**

**1.1 Introduction to Machine Learning**

**1.2 Introduction to Recurrent Neural Network**

**1.3 Objectives**

**1.4 Problem statement**

**2. LITERATURE SURVEY**

**2.1. Existing Techniques And Its Drawback**

**3. METHODOLOGY**

**3.1 Recurrent Neural Network**

**3.2 Long-Short Term Memory**

**3.3 Fundamental**

**4. IMPLEMENTATION**

**5. RESULTS**

**6. CONCLUSION**

**REFERENCES**

**CHAPTER 1**

**INTRODUCTION**

Stock markets can be defined as dynamic, unpredictable, non-linear and highly volatile in nature. Stock price predictions are very important among many business people and retail investors. Predicting stock market prices is a difficult and challenging task as they are complex and diverse and it depends on various economic factors like economic uncertainty, company’s financial reports and performance and price indicator as well as non-economic factors such as political conditions, and investor’s expectations, etc. The prices of stocks are mainly governed by demand and supply, and the ultimate goal of buying shares is to make money by buying stocks in companies whose share price is expected to jump up. Therefore to obtain higher trading profits and reduce unnecessary losses, the investors usually expect various techniques to predict and analyze the stock price movements and various trends. Stock market prediction therefore has been a major project topic among researchers in the financial area and captivates the attention of many investors. Interpreting the stock price pattern of a particular company by considering their past data and predicting their future growth and financial development will be highly beneficial.

There are two common methods of attempting to forecast stock prices of an organization. The first is fundamental analysis, which considers external factors like company profile, market situation, political and economic factors, textual-information in the form of financial news articles, social media and even blogs by economic investigators. The second is technical analysis, that attempts to find patterns in charts and use past price trends of stocks like closing and opening price, volume traded, adjacent close values and many more, to predict future price action. Now days, for predicting stock prices, advanced intelligent techniques based on either technical or fundamental analysis are used. Based on the data of historical stocks the stock price can be predicted. The most promising and prominent technique involves the use of Recurrent Neural Networks (RNN), that is basically the implementation of machine learning. Machine learning has been widely used in the capital market and plays a major role in predicting future stock prices based on historical data. Machine learning involves artificial intelligence which empower the system to learn and improve from past experiences without being programmed time and again, thereby increasing the accuracy.

Based on the data of historical stocks the stock price can be predicted. The most promising and prominent technique involves the use of Recurrent Neural Networks (RNN), that is basically the implementation of machine learning. Machine learning has been widely used in the capital market and plays a major role in predicting future stock prices based on historical data. Machine learning involves artificial intelligence which empower the system to learn and improve from past experiences without being programmed time and again, thereby increasing the accuracy.

The proposed approach considers the available historic data of a particular share and it provides predictions on a particular feature. In order to predict a share price for a required time period, the proposed model uses the time series analysis. This model applies a type of Recurrent Neural Network ( RNN ) capable of addressing linear problems and predicting time series- Long Short Term Memory (LSTM) networks. LSTM is a deep learning technique. Long-term Memory (LSTM) units execute very long sequences. LSTM evaluates the time series data by using both the historical and the present stock data accurately. LSTM replaces the traditional artificial neurons in the network layer into the most useful memory cells. With these memory cells, network is able to associate memory with remote input over time. Over the past few years, LSTM has been applied to stock market prediction in different stock markets around the world.

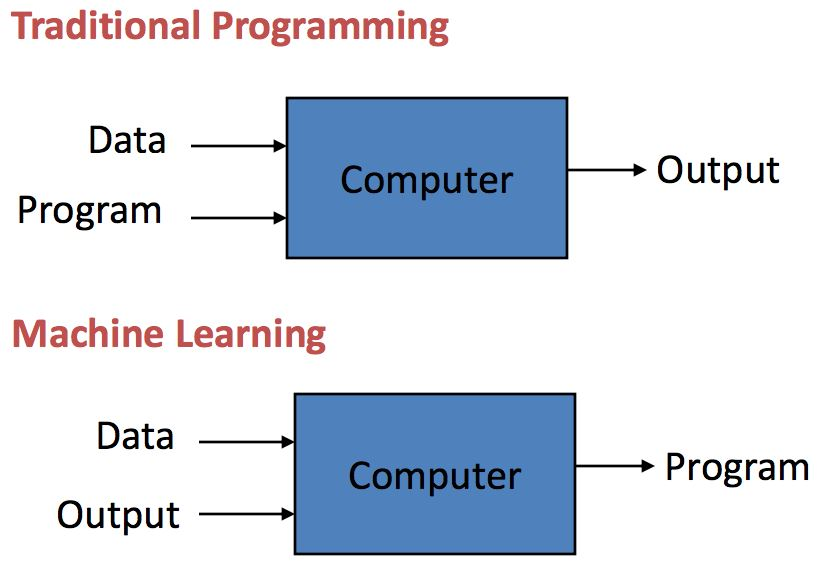
The most important aspect of machine learning is the dataset used. The dataset should be as solid and concrete as possible because a little change in the data can prolong massive changes in the results. This dataset comprises the following closing variables for companies like TATAGLOBAL, FACEBOOK, INFOSYS , TESLA,MICROSOFT and APPLE. The model is then tested with the help of test data.

* 1. Introduction to Machine Learning

Machine learning (ML) is a type of artificial intelligence ([AI](https://www.techtarget.com/searchenterpriseai/definition/AI-Artificial-Intelligence)) that allows software applications to become more accurate at predicting outcomes without being explicitly programmed to do so. Machine learning [algorithms](https://www.techtarget.com/whatis/definition/algorithm) use historical data as input to predict new output values.

Machine learning is important because it gives enterprises a view of trends in customer behaviour and business operational patterns, as well as supports the development of new products. Many of today's leading companies, such as Facebook, Google and Uber, make machine learning a central part of their operations. Machine learning has become a significant competitive differentiator for many companies.

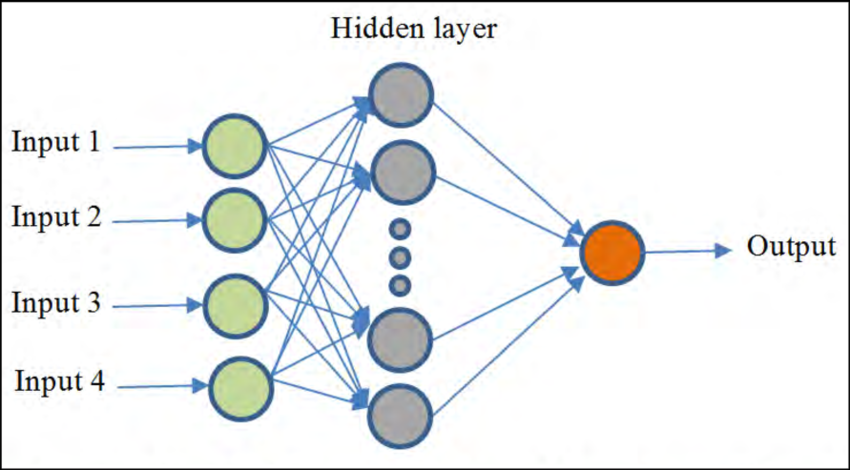
Classical machine learning is often categorized by how an algorithm learns to become more accurate in its predictions. There are four basic approaches: [supervised](https://www.techtarget.com/searchenterpriseai/definition/supervised-learning) learning**,** [unsupervised](https://www.techtarget.com/whatis/definition/unsupervised-learning) learning, semi-supervised learning and reinforcement learning. The type of algorithm data scientists choose to use depends on what type of data they want to predict.



* **Supervised learning:** In this type of machine learning, data scientists supply algorithms with labelled training data and define the variables they want the algorithm to assess for correlations. Both the input and the output of the algorithm is specified.
* **Unsupervised learning:** This type of machine learning involves algorithms that train on unlabeled data. The algorithm scans through data sets looking for any meaningful connection. The data that algorithms train on as well as the predictions or recommendations they output are predetermined.
* **Semi-supervised learning:** This approach to machine learning involves a mix of the two preceding types. Data scientists may feed an algorithm mostly labelled training data, but the model is free to explore the data on its own and develop its own understanding of the data set.
* **Reinforcement learning:** Data scientists typically use reinforcement learning to teach a machine to complete a multi-step process for which there are clearly defined rules. Data scientists program an algorithm to complete a task and give it positive or negative cues as it works out how to complete a task. But for the most part, the algorithm decides on its own what steps to take along the way.

**1.2 Introduction to Artificial Neural Networks (ANNs)**

Artificial Neural Networks (ANNs) are a fundamental concept in the field of artificial intelligence and machine learning. They are computational models inspired by the structure and functioning of the human brain. ANNs are designed to process and learn from data to perform various tasks, such as pattern recognition, classification and more.



**Key components of ANNs include:**

* **Neurons (Nodes):** In ANNs, artificial neurons, also known as nodes or units, are the basic building blocks. These neurons are interconnected in layers, typically organized into an input layer, one or more hidden layers, and an output layer. Neurons receive inputs, perform a computation, and produce an output.
* **Weights and Connections:** Each connection between neurons has an associated weight, which determines the strength of the connection. Learning in ANNs involves adjusting these weights to improve the network's performance.
* **Activation Function:**  Neurons apply an activation function to the weighted sum of their inputs to produce an output. Common activation functions include the sigmoid, ReLU (Rectified Linear Unit), and tanh functions. These functions introduce non-linearity, which is crucial for the network's ability to model complex relationships in data.

The process of using ANNs typically involves:

* **Input Data:**  The ANN receives input data, which is passed through the input layer.
* **Weighted Sum:** Each connection between neurons multiplies the input by its weight, and these weighted sums are then aggregated in each neuron.
* **Activation Function:** The aggregated sum in each neuron is passed through the activation function to produce the neuron's output.
* **Forward Propagation:** The process of moving data from the input layer through the hidden layers to the output layer is known as forward propagation.
* **Training:** ANNs learn from data through a training process, often using a technique called back- propagation. During training, the network' s weights are adjusted to minimize the difference between its predictions and the actual target values in the training data.
* **Loss Function:** A loss function measures the error between the network's predictions and the true values. The goal of training is to minimize this error.
* **Optimization:** Optimization algorithms, such as gradient descent, are used to update the weights iteratively and reduce the loss function.

ANNs have found applications in a wide range of fields, including image and speech recognition, natural language processing, autonomous vehicles, financial modelling, and many more. Deep Learning, a subset of machine learning, involves the use of deep neural networks with multiple hidden layers, and has achieved remarkable success in various complex tasks.

Artificial Neural Networks have revolutionized the field of machine learning, making them a powerful tool for solving problems that involve complex patterns and large datasets.

**1.3 Introduction to Recurrent Neural Networks (RNNs)**

In the ever-expanding topic of deep learning and artificial intelligence (AI), Recurrent Neural Networks (RNNs) have emerged as a powerful and versatile class of neural networks. Unlike traditional feed-forward neural networks, RNNs are designed to process sequences of data, making them exceptionally well-suited for a wide array of tasks involving sequential or time-dependent information. Whether it's natural language processing, speech recognition, time series analysis, or beyond, RNNs have proven to be invaluable tools for modeling and understanding sequential data.

At the heart of an RNN's design is its ability to maintain a form of memory, enabling it to capture and exploit temporal dependencies in the data it processes. This memory capacity, often referred to as "hidden state" or "context," equips RNNs with the ability to remember information from past time steps, making them ideal for tasks where the order of data elements matters. Consequently, RNNs can seamlessly adapt to varying - length input sequences, offering a level of flexibility and versatility that distinguishes them from their feed-forward counter parts.

In this introduction to RNNs, we will look into the fundamental principles that underpin their operation, exploring the architecture, training process, and the challenges they aim to address. We will also examine common applications of RNNs, highlighting their impact on fields like natural language understanding, speech generation, and financial prediction. Moreover, we will touch upon some of the limitations of traditional RNNs and the innovations that have evolved to overcome these challenges, such as Long Short-Term Memory (LSTM) and Gated Recurrent Unit(GRU) networks. As we navigate the intricate world of RNNs, we'll uncover their pivotal role in modelling sequential data, setting the stage for a deeper exploration of this influential branch of deep learning.

* 1. **Objectives**

Predicting stock returns and a company's financial situation in advance will provide greater benefits for investors in the current competitive market, all wing them to invest with confidence. Stock forecasting can be done with the help of current and historical market data.

This project aims to investigate and develop supervised learning systems for stock price prediction. It must calculate the stock's projected price using past data. It should also be able to visualize the market index in real time.

The goal of stock market prediction is to predict the future movement of a financial exchange's stock value. Investors will be able to make more money if they can accurately predict share price movement.

* 1. **Problem Statement**

The stock market appears in the news every day. You hear about it every time it reaches a new high or a new low. The rate of investment and business opportunities in the Stock market can increase if an efficient algorithm could be devised to predict the short - term price of an individual stock. Previous methods of stock predictions involve the use of Artificial Neural Networks and Convolution Neural Networks which has an error loss at an average of 20%. In this project, we will see if here is a possibility of devising a model using Recurrent Neural Network which will predict stock price with a less % of error. And if the answer turns to be YES, we will also see how reliable and efficient will this model.

**CHAPTER 2**

**LITERATURE SURVEY**

In today's dynamic financial landscape, predicting stock prices has become a critical challenge, impacting investors, analysts, and researchers. The stock market's volatility and susceptibility to external factors have elevated the importance of accurate price forecasting. This literature review explores the various methods and approaches used in the pursuit of stock price prediction, highlighting progress, challenges, and future research possibilities. Stock price prediction is pivotal for informed investment decisions and economic stability. This review outlines the historical evolution of predictive techniques, from basic statistical models to advanced machine learning algorithms. It also addresses the complex determinants of stock price movements, considering financial indicators, market sentiment, and external events.

Despite advancements, challenges like the efficient market hypothesis and data issues persist. This literature review delves into ethical concerns, model limitations, and the potential for market manipulation. By synthesizing existing research, it provides a comprehensive understanding of the stock price prediction landscape, fostering better-informed financial decisions and a deeper grasp of market dynamics**.**

The proposed work by Pritam Ahire, Hanikumar Lad, Smita Parekh, Saurabh Kabrawala, D.Y Patil Institute of Engineering and Technology, Pune, India focuses on the use of recurrent neural network (RNN) based Machine learning techniques known as Long Short Term Memory (LSTM) to predict stock values and to provide efficient stock price prediction. The system predicts the stock prices for companies like Alcoa Corp Company, Carnival Corp, Tesla Corp. etc. The system is based on five factors such as: Date of stock price, Opening price, High, Low, Volume and the Close Interest for the respective companies.

The system presents a new model for optimizing stock forecasting. The authors Ya Gao, Rong Wang , and Enmin Zou from School of Public Finance and Taxation, Central University of Finance and Economics, Beijing, China, School of Computer Science and Technology, Xidian University, Xi’an, China and School of Electronics and Information, Xi’an Jiaotong University, Xi’an, China include a range of technical indicators, including investor sentiment indicators and financial data, and perform dimension reduction on the many influencing factors of the retrieved stock price using deep learning approaches as LASSO and PCA. In addition to this, a comparison of the performance of LSTM & GRU for stock market forecasting under various parameters is performed. The result of which show that both the approaches- LSTM and GRU can equally predict the stock prices efficiently.

The authors Adil Moghar and Mhamed Hamiche from University Abdelmalek Essaad, Morocco have build a model to see in which precision a Machine learning algorithm can predict and how much the epochs can improve the  model. The system is build using Long-Short Term Memory model (LSTM) technique. The data in this paper consist the daily opening prices of two stocks in the New-York Stock Exchange NYSE (GOOGLE and NKE) extracted from yahoo finance, for GOOGLE. For training the data the system uses mean squared error to optimize the model. Also, different Epochs for training data are used. This model is capable of tracing the evolution of opening prices for both assets used.

The proposed system is presented by Pramod B S and Mallikarjuna Shastry P. M, from REVA University, Bengaluru. As compared to today’s available prediction algorithms, this system gives accurate results. The proposed algorithm uses machine learning techniques like Recurrent Neural Networks named as Long Short Term Memory using the market data to predict the share price. The network is trained and evaluated with multiple sizes of input data urge the graphical outcomes. The various parameters used in this model comprise of Date, Open, Close, Volume / trade quantity, High, Low and Turnover of TATAMOTORS share. The proposed model is able to predict the share price with very low loss and error rate if there is an increase the epoch batch rates, the training will be more methodical.

This paper focuses on the usage of Regression and LSTM based Machine learning predict stock values. Regression and LSTM models are engaged for this speculation separately. Regression involves minimizing the errors and LSTM remembers the data and results for the long run. The authors in this project, Ishita Parmar, Navanshu Agarwal, Sheirsh Saxena, Ridam Arora, Shikhin Gupta, Himanshu Dhiman, Lokesh Chouhan, Department of Computer Science and Engineering, National Institute Of Technology, Hamirpur, India, uses supervised machine learning on a dataset obtained from Yahoo Finance. This dataset comprises the following five factors: open, close, low, high and volume.

The proposed system by authors Raghav Nandakumar, Uttamraj K R, Vishal R,Y.V Lokeswari, Department of Computer Science and Engineering, SSN College of Engineering, Chennai, Tamil Nadu, India, propose an online learning algorithm which uses recurrent neural network(RNN) called Long Short Term Memory(LSTM), where the weights are adjusted for individual data points using stochastic gradient descent. This provides accurate outcomes when compared against existing stock price prediction algorithms. Benchmark stock market data was obtained from two primary sources: Yahoo Finance and Google Finance. The obtained data contained five features: Date, Opening price, High, Low, Volume, OpenInt. Also a comparison with respect to accuracy is done against an Artificial Neural Network. The accuracy of the prediction model is calculated using the RMSE (Root Mean Squared Error) metric.

The research work done by V Kranthi Sai Reddy Student, ECM, Sreenidi Institute of Science and Technology, Hyderabad, India. In the finance world stock trading is one of the most important activities. Stock market prediction is an act of trying to determine the future value of a stock other financial instrument traded on a financial exchange. This paper explains the prediction of a stock using Machine Learning. The technical and fundamental or the time series analysis is used by the most of the stock brokers while making the stock predictions. The programming language is used to predict the stock market using machine learning is Python. In this paper we propose a Machine Learning (ML) approach that will be trained from the available stocks data and gain intelligence and then uses the acquired knowledge for an accurate prediction. In this context this study uses a machine learning techniques called Support Vector Machine(SVM) to predict stock prices for the large and small capitalizations and in the three different markets, employing prices with both daily and up-to-the-minute of frequencies.

Forecasting the Stock Market Index Using Artificial Intelligence Techniques this research work done by Lufuno Ronald Marwala , A dissertation submitted to the Faculty of Engineering and the Built Environment, University of the Witwatersrand, Johannesburg, in fulfillment of the requirements for the degree of Master of Science in Engineering. The weak form of Efficient Market hypothesis (EMH) states that it is impossible to forecast the future price of an asset based on the information contained in the historical prices of an asset. This means that the market behaves as a random walk and as a result makes forecasting impossible. Furthermore, financial forecasting is a difficult task due to the intrinsic complexity of the financial system. The objective of this work was to use artificial intelligence (AI) techniques to model and predict the future price of a stock market index. Three artificial intelligence techniques, namely, neural networks , support vector machines and neuro-fuzzy systems are implemented in forecasting the future price of a stock market index based on its historical price information’s. Artificial intelligence techniques have the ability to take into considered on financial system complexities and they are used as financial time series forecasting tools. Two techniques are used to benchmark the AI techniques, namely, Auto regressive Moving Average (ARMA) which is linear modeling technique and random walk (RW) technique. The experimentation was performed on data obtained from the Johannesburg Stock Exchange. The data used was a series of past closing prices of the All-Share Index. The results showed that the three techniques have the ability to predict the future price of the Index with an acceptable accuracy. All three artificial intelligence techniques out-performed the linear model. However, the random walk method out-performed all the other techniques. These techniques shows an ability to predict the future price however, because of the transaction costs of trading in the market, it is not possible to show that the three techniques can disprove the weak form of market efficiency. The results show that ranking of performances support vector machines, neuro- fuzzy systems, multilayer perceptron neural networks is dependent on the accuracy measure used.

**2.1. Existing Techniques And Its Drawback**

As it is mentioned in the introduction, traditional approaches to stock market prediction and analysis includes Fundamental and technical analysis. Fundamental analysis dwells into stocks past performance, company’s credibility, and other factors like news, economy. It uses two fundamental research indicators: P/E ratio and P/B ratio for long-term price fluctuations. Whereas technical analysis dives into investing at trends and employs price charts and formulae to anticipate future stock values.

Fundamental analysis can become tedious and time consuming. It has drawbacks of extrapolation, time delay, need for long term investment, and has no reliable trade’s signals. Fundamental analysis' drawback is that it can bring you on board a good stock at the wrong time, requiring you to hold on to the stock for a prolonged time.

Technical analysis has limitations like any other method.  It's possible to misread the graph. It's possible that the formation is based on low volume. The moving average periods employed may be too long or too short for the type of transaction you're attempting to make. For the same stock, one technical analysts view may differ from that of another. The technical approaches employed by analysts to examine equities can differ from one to the next.

**CHAPTER 3**

**METHODOLOGY**

By proper use of machine learning techniques and algorithms, we can relate the previous data to the current data and train the machine to learn from it and make proper assumptions. Machine learning has many models but in this project focuses one of the most important and accurate of them which make the predictions efficient using it. The project will be a great asset for traders and investors for investing money in the stock market since it is trained on a huge collection of historical data. The project demonstrates the use of a machine learning model i.e. LSTM to predict the stock value with more accuracy as compared to other models. Analysis of stocks using deep learning will be useful for new investors to invest in the stock market. Stock market includes daily activities like nifty and sensex calculation, exchange of shares.

The proposed system measures the accuracy of stock prices by using the predictions for the test set and the actual values. The system also uses different areas of research including data pre-processing, LSTM and so on. In this proposed system, we will focus on predicting the future trends of the stock values using machine learning algorithms Long-Short Term Memory (LSTM) algorithm, a type of recurrent neural network. In this system, train the machine by taking the various datasets from the past to make an accurate future prediction. Datasets from previous 10 years' stocks are used to train the model. In this project will majorly use five libraries like numpy, panda, DateTime, Matplotlib and scikit to solve the problem.

The system works on a Yahoo Finance, which has a record of all the dates and its raw data of the closing variable. From this raw data, knowledge is extracted by performing data pre-processing and refining to predict a close information for requested date of future. Once the knowledge is available, it will be fed to the LSTM algorithm to perform stock prediction and give a data visualization using python, this investment prediction will be subdivided into different time frames and a suitable advice from the prediction can be given to the consumer, as shown below.

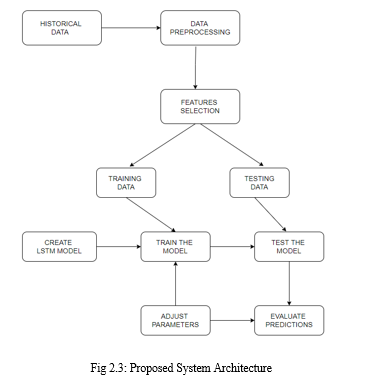
****

Fig: Proposed System Architecture

In order to predict the future of the stock market, a precise approach must be followed, as shown in this diagram.

* The first phase will be to gather historical stock data for any company, which will be used to forecast stock prices.
* The data is then pre processed using techniques such as data scaling and data discretization.
* Only the features that will be supplied to the neural network are chosen in this step: date, open, high, low, close, and volume.
* In a 80:20 ratio, historical stock data is separated into training data and testing data.
* The data is placed into a recurrent neural network, which is then trained to make predictions. A sequential input layer is followed by LSTM layers, a dense layer, and finally a dense output layer with linear activation function in LSTM model.
* The target value is compared to the output value generated by the RNN's output layer. The back propagation through time algorithm, which modifies the weights and biases of the network, minimizes the error between the target and the acquired output value.

**3.1 Recurrent Neural Network (RNN)**

Recurrent Neural Network (RNN) is a type of [Neural Network](https://www.geeksforgeeks.org/tag/neural-network/) where the output from the previous step is fed as input to the current step. A recurrent neural network (RNN) is a type of artificial neural network which uses chronological data or time series data. Recurrent neural networks (RNN) utilize training data for knowledge. They are distinguished by their “memory” as they take information from previous inputs to influence the current input and output. On the other hand, traditional neural networks assume that inputs and outcomes are not related to each other, the output of the recurrent network depends on the prior attributes within the sequence. The main and most important feature of RNN is Hidden state, which can remember some data about a sequence.

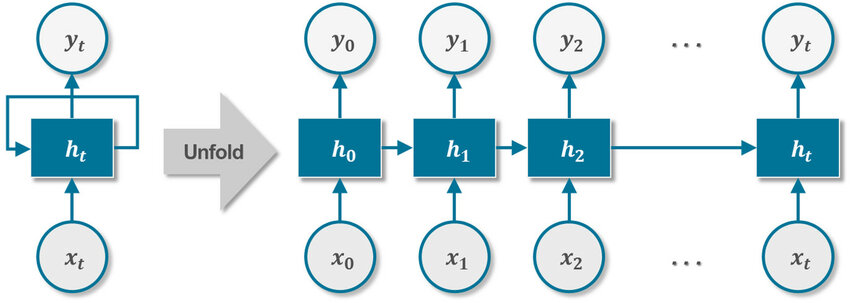


Fig: Recurrent Neural Network

Simple Neural Networks also learn and retain what they've learned, which is how they predict classes or values for fresh datasets. However, unlike conventional Neural Networks, RNNs rely on information from prior output to predict forthcoming data/input. When dealing with sequential data, this capability comes in handy. LSTM is the most commonly used RNN model.

Therefore, for stock prediction RNN is used over ANN.

**3.2 Long-Short Term Memory (LSTM)**

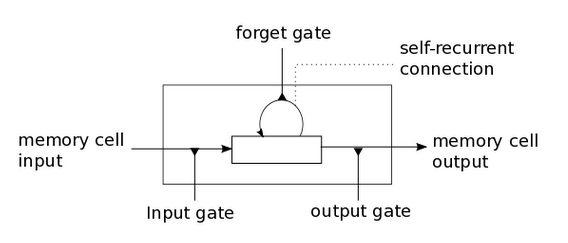
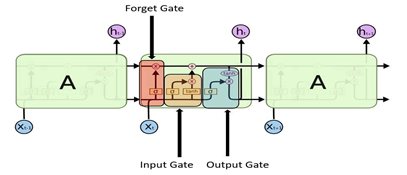
****

Fig: Long Short Term Memory

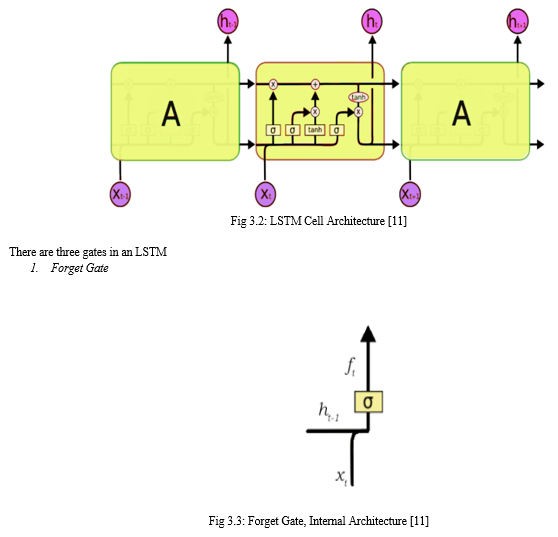
* The LSTM (Long Short Term Memory) neural network is a form of neural network that is particularly useful in time series forecasting.
* Hochreiter and Schmidhuber first proposed long short-term memory in 1997 to overcome the aforementioned issues.
* By incorporating memory cells and gate units into the neural network design, long-short term memory addresses the difficulty of learning to recall information over a time interval.
* In a traditional neural network, final outputs are rarely used as an output for the following phase, but if we look at a real-world example, we can see that our final output is often dependent not just on external inputs but also on earlier output.
* When humans, for example, read a book, each sentence's comprehension is based not only on the current list of words, but also on the prior sentence's comprehension or the context provided by previous sentences. Humans don't have to start again every time they think.
* RNN can remember long-term inputs thanks to LSTM. Similar to computer memory, it stores information in memory.
* It has the ability to read, write, and delete data from its memory. This memory can be thought of as a closed cell with a closed description that decides whether to save or remove data.

**A.LSTM Architecture**

****

**Fig: LSTM Cell Architecture**

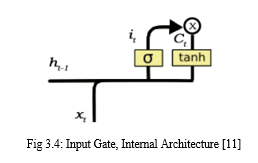
1. **Forget Gate**



**Fig: Forget Gate**

* A forget gate is responsible for removing information from the cell state.
* Information that is no longer necessary for the LSTM to understand things or that is of lesser importance is removed.
* The forget gateway governs when newer information will be introduced into certain areas of the cell.
* It produces values that are near to 1 for sections of the cell state that should be kept and zero for values that.
* This gate takes in two inputs; h\_t-1 and x\_t.
* h\_t-1 is the hidden state from the previous cell or the output of the previous cell and x\_t is the input at that particular time step.

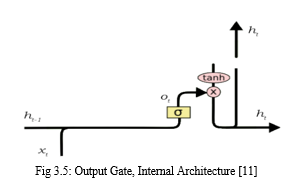
1. **Input Gate**



**Fig: Input Gate**

* The input gate is responsible for the addition of information to the cell state.
* Regulating what values need to be added to the cell state by involving a sigmoid function. This is basically very similar to the forget gate and acts as a filter for all the information from h\_t-1 and x\_t.
* Creating a vector containing all possible values that can be added (as perceived from h\_t-1 and x\_t) to the cell state. This is done using the tanh function, which outputs values from -1 to +1.
* Multiplying the value of the regulatory filter (the sigmoid gate) to the created vector (the tanh function) and then adding this useful information to the cell state via addition operation.

1. **Output Gate**

****

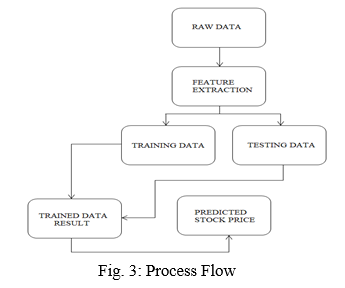
**Fig: Output Gate**

* The output gate is responsible for selecting useful information from the current cell state and showing it out as an output.
* An output gate's operation can be broken down into three parts once more:
* Creating a vector after applying tanh function to the cell state, thereby scaling the values to the range -1 to +1.
* Making a filter using the values of h\_t-1 and x\_t, such that it can regulate the values that need to be output from the vector created above. This filter again employs a sigmoid function.
* Multiplying the value of this regulatory filter to the vector created in step 1, and sending it out as an output and also to the hidden state of the next cell.

**Advantages of LSTM**

* The ability of LSTM to read intermediate context is its key advantage.
* Without explicitly applying the activation function inside the recurring components, each unit recalls facts for a long or short length of time.
* The release of the forget gate, which ranges between 0 and 1, is the sole way for any cell state to be repeated.
* To put it another way, the LSTM cell's forgetting gateway is in charge of both the hardware and the function of cell state activation.
* As a result, instead of explicitly increasing or decreasing in each step or layer, the data from the preceding cell can flow through the unmodified cell, and the instruments can convert to their suitable values over a limited time.
* Because the amount stored in the memory cell is not transformed in a recurrent fashion, the gradient does not cease when trained to distribute rearward, allowing LSTM to solve a perishable gradient problem.

**3.3 Fundamentals**

****

**Fig: Data Flow Diagram**

* **Data Collection:**  Data collection is the basic and initial step. It deals with the collection of the right dataset. Based on various aspects, the dataset that is to be used in the stock market prediction has to be modified. Data collection also complements the dataset by adding more data that is external. The data used in this project mainly consists of the previous few years stock prices. Initially, we will be analysing the data from Yahoo Finance and according to the accuracy, we will be using the model with the data to analyse the predictions accurately.
* **Pre Processing:**  Data pre-processing is a part of data mining, which involves transforming raw data into a more reasoned format. Raw data is usually incomplete or inconsistent and usually contains many errors. The data pre-processing involves checking out for missing values, looking for categorical values, splitting the data-set into training and test sets and finally doing a feature scaling to limit the range of variables so that they can be compared on common technologies .
* **Training the Machine:** Training the machine is similar to feeding the data to the algorithm to test the data. The models are tuned and fitted using Training sets. The training of the model comprises cross-validation where we get a well-grounded approximate performance of the model using the training data.
* **Data Scoring:** Scoring the data is referred to as a process of applying a predictive model to a set of data. The technique used to process the dataset is the Long-Short Term Memory. We achieve interesting results, based on these learning models. Thus describes how the result of the model can help to predict the possibility of a stock to rise and fall based on certain parameters
* **Output Block Description:** The stock selected by the user acts as an input to the system. The selected stock contains the time period for which the user needs the prediction. Analysis of the input data takes place resulting in generation of graph which acts as an output.

**CHAPTER 4**

**IMPLEMENTATION**

**I. Importing Libraries**

Let's begin by importing some library numpy, pandas from, pandas data reader, and min max color from sklearn preprocessing. Next,step is to import sequential from keras models, and Dense, LSTM from keras layers. Finally, i'll import matplotlib.

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns

from pandas\_datareader.data import DataReader

import yfinance as yf

from pandas\_datareader import data as pdr

from datetime import datetime

**II. Plot Close Price Movement**

I’m going to plot a figure here. I'll pay more attention to the close price because that's the price we'll predict.

plt.figure(figsize=(16,6))

plt.title('Close Price History')

plt.plot(df['Close'])

plt.xlabel('Date', fontsize=18)

plt.ylabel('Close Price USD ($)', fontsize=18)

plt.show()

**III. Feature Selection**

In this prediction will be in the closing price, so create a new data frame with only the close price column using the filter function. Also because next step is to normalization the data later, use the value function to convert all off our data frames to numpy array.

data = df.filter(['Close'])

dataset = data.values

length(dataset)

**IV. Training Data**

Prepare the traing data so that we can use 80% of the data in training, so I will use the mat library and the mat function to create training data size. Receives the 80% of the data for training.

data = df.filter(['Close'])

dataset = data.values

training\_data\_len = int(np.ceil( len(dataset) \* .8 ))

training\_data\_len

**V. Data Normalization**

Here I have Normalized and transformed the training data. Normalization is the process of rescaling data from its original range so that all values fall between 0and 1. With the help of MinMaxScaler i normalizedour data. It is imported from sklearn.preprocessing module which includes scaling, centering, normalization, binarization methods.

scaler = MinMaxScaler(feature\_range=(0,1))

scaled\_data = scaler.fit\_transform(dataset)

scaled\_data

**VI. Scaling Train Data**

I'll make a training dataset with values ranging from zero to data size, which will account for 80% of our training dataset.

Next, I'll divide the training data into two groups: x train and y train.

The independent training variable is the x train, while the dependent or target variable is the y train.

I need 60 observations of our training data, so I'll make a loop for e that's in the range of 60 to the size or total size of the training data. I'll append the plus 60 values to the xtrain data, making the x train data start at position i-60 and they train data contain the value in the 61st position. As a result, x will have 60 values ranging from 0 to 59, while y will have 61 values and the 60th position. This is the value I want to predict with our model.

train\_data = scaled\_data[0:int(training\_data\_len), :]

x\_train = []

y\_train = []

for i in range(60, len(train\_data)):

x\_train.append(train\_data[i-60:i, 0])

y\_train.append(train\_data[i, 0])

if i<= 61:

print(x\_train)

print(y\_train)

print()

**VII. Train Data Reshaping**

First of all, the x\_train and y\_train were converted into numpy arrays which were used to tutor the LSTM.

Because the LSTM layer is a recurrent layer, it anticipates a three-dimensional input. For that purpose we reshaped the data in 3D as (1670, 60, 1).

1670 = Number of Samples

60 = Number of steps

1 = Number of Features

x\_train, y\_train = np.array(x\_train), np.array(y\_train)

x\_train = np.reshape(x\_train, (x\_train.shape[0], x\_train.shape[1], 1))

x\_train.shape

**VIII. Building LSTM Model**

Process of building LSTM start with calling sequential () which is imported from keras model. Further on, we have inserted one LSTM layer including 50 neurons and two dense layers with 25 and 1 neurons respectively. The input shape contains a number of steps and features.

For compiling this model I`m used an 'adam' optimizer.

An optimizer adjusts the neural network's properties like weights and learning rate. As a result, it aids in the reduction of total loss and the improvement of accuracy. LSTM models are trained by calling the fit() function. I have usedtrain data to fit the LSTM model with 1epochs and used a batch size of 1.

The number of epochs is the number of complete passes through the training dataset.

Batch size refers to the number of training examples utilized in one iteration

from keras.models import Sequential

from keras.layers import Dense, LSTM

model = Sequential()

model.add(LSTM(128, return\_sequences=True, input\_shape= (x\_train.shape[1], 1)))

model.add(LSTM(64, return\_sequences=False))

model.add(Dense(25))

model.add(Dense(1))

model.compile(optimizer='adam', loss='mean\_squared\_error')

model.fit(x\_train, y\_train, batch\_size=1, epochs=5,verbose=2)

**IX. Testing Data**

Test data is used to test the model here. I have the remaining 20% data which is test data. An array that contains 20% normalized data is created. 60 observations of testing data are created here which will be handy for prediction. y\_test is the actual data here and x\_test will be the predicted data.

So, actual data and predicted data will be compared later on.

test\_data = scaled\_data[training\_data\_len - 60: , :]

x\_test = []

y\_test = dataset[training\_data\_len:, :]

for i in range(60, len(test\_data)):

x\_test.append(test\_data[i-60:i, 0])

**X. Converting Test Data**

Then we turned the test data to an array and used the reshape tool to reshape the data to three dimensions so that we could use it in a prediction model. Now, i've used the model to build a prediction from our testing data and then we've used the scaler that inverse transforms to get the prediction's value.

x\_test = np.array(x\_test)

x\_test = np.reshape(x\_test, (x\_test.shape[0], x\_test.shape[1], 1 ))

predictions = model.predict(x\_test)

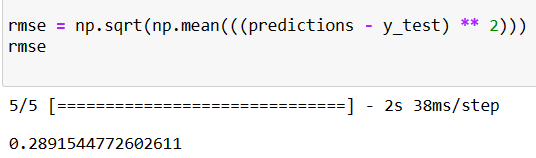
predictions = scaler.inverse\_transform(predictions)

**CHAPTER 5**

**RESULTS**

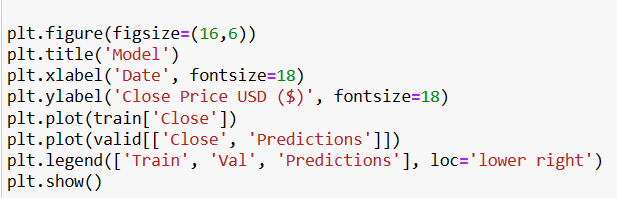
**A. Root Mean Square Error**

To evaluate the model, the root mean square error (rmse) will be calculated, which will tell us how accurate the model is. The rmse is 0.2 , which is a very good figure because it indicates that the model is very accurate and that the forecast is extremely near to the actual value.

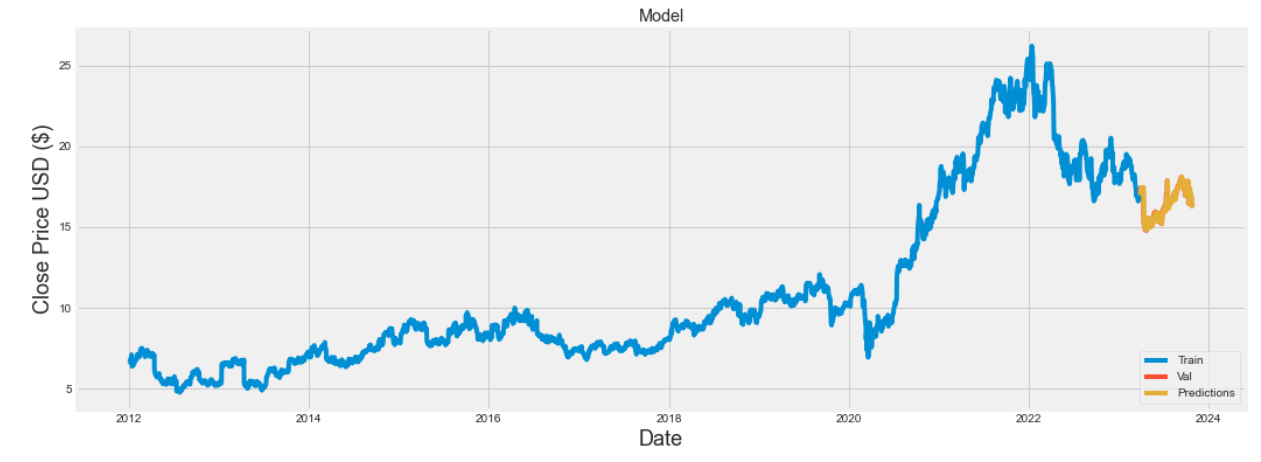


**B. Predicted Data**

Next plotted the data to see how close the prediction to the actual values.

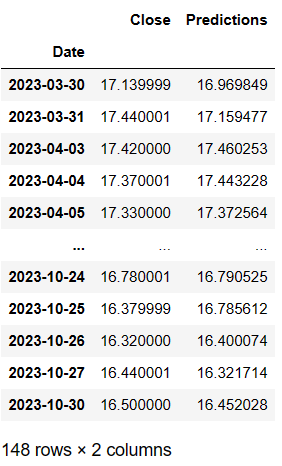


As we can see, the blue line represents the training data, which accounts 80% of the green line represents the prediction, while the Orange line represents the actual value. As you can see, the predicted values and the actual values are nearly identical, so now use this model to predict any value we want and enable any stocks value we want.



**C. Values**

We can see that the closing price and prediction are really near, with the exception of a few dates where they deviate somewhat, but in general they are both very close.



**CHAPTER 6**

**CONCLUSIONS**

Predicting the stock market is a time-consuming and strenuous procedure. However, with the introduction to Machine Learning and its various algorithms, the Stock Market Prediction advancements have begun to include such approaches in analysing stock market data. By measuring the accuracy of the different algorithms, found that the most suitable algorithm for predicting the market price of a stock based on various data points from the historical data is the Long-Short Term Memory (LSTM) algorithm. The algorithm will be a great asset for brokers and investors for investing money in the stock market since it is trained on a huge collection of historical data and has been chosen after being tested on a sample data.

The project demonstrates the machine learning model to predict the stock value with more accuracy as compared to other machine learning models. In the future, the stock market prediction system can be further improved by utilizing a much bigger dataset having higher computing capacities than the one being utilized currently and number of training epochs that better suit our assets and maximize our predictions accuracy. Furthermore, other models of Machine Learning could also be studied to check for the accuracy rate resulted by them. It has led to the conclusion that it is possible to predict the stock market with more accuracy and efficiency using machine learning techniques.

Even though good scores are achieved using ML algorithms, there can be an improvement. Adding more data helps the algorithm to learn better. Hyper parameter optimization is another method of tuning the hyper parameters to get the best performance on the dataset provided. It can be implemented using the Scikit-learn machine learning library. Deep Learning algorithms can be implemented to predict accurate results. Deep learning is a branch of machine learning where neural networks algorithms are inspired by the human brain. Long-short term memory (LSTM) can be implemented to predict the stock price. LSTM can learn order dependence in sequence prediction problems. Artificial Neural Network (ANN) is also an extremely recognized method for predictive finance. ANNs are multi-layer fully connected neural nets. Convolutional Neural Networks (CNN) are made up of neurons with biases and learnable weights. CNNs, which are designed to map image data to an output variable, can help to improve predictions.

The future prospects include building a Machine learning web app in Python where the user can simply input a stock dataset and get appropriate output with the highest accuracy. The machine app should take in the dataset correctly and choose the algorithm that gives the lowest error rate. The predictions should get printed on the screen. The user interface should be easy and user-friendly for beginners. The app can then be deployed on servers like Heruko to see the model in action.

**REFERENCES**

1)Pritam Ahire, Hanikumar Lad, Smit Parekh, Saurabh Kabrawala, “LSTM Based Stock Prediction,” International Journal of Creative Research Thoughts(IJCRT), vol. 9, pp. 5118-5122, Feb. 2021.

2) Ya Gao, Rong Wang, and Enmin Zou, “Stock Prediction Based on Optimized LSTM and GRU Models,” Hindawi, vol. 2021, pp. 1-8, Sept. 2021.

3) Adil Moghar and Mhamed Hamiche, “Stock Market Prediction Using LSTM Recurrent Neural Network,” Science direct, vol. 170, pp.1168-1173, Apr. 2020.

4) Mallikarjuna Shastry P. M. and Pramod B S, “Stock Price Prediction Using LSTM,” ResearchGate, vol. 83, pp. 5246-5251, May. 2020.

5) Ishita Parmar, Navanshu Agarwal, Sheirsh Saxena, Ridam Arora, Shikhin Gupta, Himanshu Dhiman, Lokesh Chouhan, “Stock Market Prediction Using Machine Learning,” in First Inter- national Conference on Secure Cyber Computing and Communications, 2018, doi: 10.1109/ICSCCC.2018.8703332

6) Uttamraj K R, Raghav Nandakumar, Vishal R, Y.V Lokeswari, “Stock Price Prediction Using Long Short Term Memory,” International Research Journal of Engineering and Technology (IRJET), vol. 05, pp. 3342-3348, Mar. 2018.

7) Gareja Pradip, Chitrak Bari, J. Shiva Nandhini, "Stock market prediction using machine learning" International Journal of Advance Research andDevelopment, Volume 3, Issue 10, 2018

8) K. Raza, "Prediction of Stock Market performance by using machine learning techniques", 2017 International Conference on Innovations in Electrical Engineering and Computational Technologies (ICIEECT),

9) K. Hiba Sadia, Aditya Sharma, Adarrsh Paul, Sarmistha Padhi, Saurav Sanyal, "Stock Market Prediction Using Machine Learning Algorithms", International Journal of Engineering and Advanced Technology (IJEAT), Volume-8 Issue-4, 2019

10) Raut Sushrut Deepak, Shinde Isha Uday, Dr. D. Malathi, "MACHINE LEARNING APPROACH IN STOCK MARKET PREDICTION", International Journal of Pure and Applied Mathematics, Volume 115, No. 8, 2017.

11) Mehtab, S., Sen, J., A robust predictive model for stock price prediction using deep learning and natural language processing. In: Proceedings of the 7thInternational Conference on Business Analytics and Intelligence

12) M. Usmani, S. H. Adil, K. Raza and S. S. A. Ali, "Stock market prediction using machine learning techniques", 2016 3rd International Conference onComputer and Information Sciences (ICCOINS)

13) Tang, J., Chen, X., Stock market prediction based on historic prices and news titles. In: Proceedings of the International Conference on Machine Learning Technologies (ICMLT)

14) Ashish Sharma, Dinesh Bhuriya, Upendra Singh. "Survey of Stock Market Prediction Using Machine Learning Approach", ICECA 2017

15) Sachin Sampat Patil, Prof. Kailash Patidar, Asst. Prof. Megha Jain, “A Survey on Stock Market Prediction Using SVM”, IJCTET 2016.