National College of Ireland

Short-Term Price Movement Prediction for Stock Market Indexes and its Application

MSc Research Project
Data Analytics

Tejveer Singh Goraya
Student ID: 19202687

School of Computing
National College of Ireland

Supervisor: Dr. Bharathi Chakravarthi

National College of Ireland
Project Submission Sheet
School of Computing

Student Name:	Tejveer Singh Goraya
Student ID:	19202687
Programme:	Data Analytics
Year:	2021
Module:	MSc Research Project
Supervisor:	Dr. Bharathi Chakravarthi
Submission Due Date:	16/12/2021
Project Title:	Short-Term Price Movement Prediction for Stock Market Indexes and its Application
Word Count:	6,377
Page Count:	21

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:	Tejveer Singh Goraya
Date:	15th December 2021

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:

Attach a completed copy of this sheet to each project (including multiple copies).	☐
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	☐
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	☐

Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Short-Term Price Movement Prediction for Stock Market Indexes and its Application

Tejveer Singh Goraya
19202687

Abstract

Machine Learning algorithms have been applied to a wide variety of applications in the domain of finance. One such application of ML is predicting future prices/values of stocks and indexes listed in the stock market. Primarily, this research will make an attempt at predicting stock prices/index values for a period 30 days in the future.Secondly,this research will explore a niche in the market that has been slightly overlooked due to its feasibility in the financial derivatives domain and demonstrate an application for the same. This research will shed a light on two main types of financial derivatives(Call and Put Options) and how machine learning will help in predicting the price of these financial derivative over a period of 30 days(As these financial products are only valid for 30 days). By taking advantage of the large amount of literature available in the stock market prediction domain, this research will aim to predict the stock prices/index values over a period of 30 days by modelling and testing several types of Machine learning algorithms ranging from LSTM,GRU and CNN including each of their types. The final outcome of this study is to accurately predict the stock prices/index values for a short period of time(30 days) and implement this information in order to select an appropriate Call/Put option with a help of a Simulation technique(Monte Carlo Simulation and its type) in order to yield a maximum profit in the market.

1 Introduction

A country's financial health can be fairly judged by its financial markets. Researchers have studied and explored multiple machine learning and statistical models in order to predict the future movements of stock market prices and index values. Theoretically these models have been successful in predicting the values over a long period of time, but the financial markets are heavily influenced by uncertainties such as Policy changes, Geo-political instability, War and Acts of God (Greenwood and Smith (1997)). To overcome these uncertainties of the long term predictions, an improvised approach of short term price and value prediction of stock market has been taken into consideration in this research. Apart from these short term predictions an application of these have also been explored in this research.

Do machine learning algorithms predict short term movements(30 days) of stock market as efficiently as long term movements? There is vast amount of research and data available on stock market prediction ranging from basic Vanilla networks to highly complicated Neural Networks. Each of these machine learning models

have a specific function and they fit into a particular requirement. This research aims to shed a light onto the available methods by exploring and applying majority of the models in order to find the best fit for the application of this research.

In order find the model which perfectly fits into the application and narrow down the search, this research takes into consideration the models which are efficient in accurately predicting long term stock market prices and index values. The long term range is considered to be anywhere between 6 months to 2 years. As stock market is highly unpredictable in the long run due to uncertainties such as market crash,natural disasters and policy changes, it is a more viable option to predict the stock market and indexes for a period of 1 month or 30 days which will minimise the probability of uncertainties which exist in the long run (De Roon et al. (1996)).Another reason for choosing this time period is the application which will be explained in the later sections. Furthermore, to reduce the margin of error the period of time (the year and month under consideration - 2016 to 2017 and 2021) are chosen by using *K-means clustering for outliers identification* and multiple cases within this are discussed below as well.

The primary challenge while doing this is to work with less amount of data available for testing and training the model and maintaining the accuracy at the same time. To accommodate for these challenges two main types of ML models are being studied namely*LSTM and CNN*, further these models are designed to accommodate variations which include *Bidirectional,Sequence 2 Sequence, 2 Path and variational autoencoder(Vae).*After evaluating multiple models,this research will demonstrate an application of the appropriate models in the financial derivatives market by identifying appropriate Call/Put options for investors. This application is accomplished with the help of simulations in order to apply these models to live data.

2 Related Work

This section explores and discusses the available literature in line with this research and breaks it down into two parts. The first part looks into the ML algorithms and models that are used to predict stock prices for longer time periods and have relatively superior results. The later part explores the literature to identify key gaps in the domain and solidify the need for this research in the financial derivatives domain.

2.1 Machine Learning in Stock prediction:

This subsection explores the available literature about three ML models namely LSTM,GRU and CNN along with their applications in the domain of long term stock market predictions. The range of data that is used in studies that in the stock market prediction domain are usually in the range of 5-10 years as this gives a dataset of more than around 1500-3000 data points(Vijh et al. (2020)).

2.1.1 LSTM - Long short-term memory

Predicting the prices of stocks listed on the Chinese stock market using data from 15 years in the past, yields results which are highly efficient using the simple LSTM model. The accuracy achieved in this case is around 27.1%. This phenomenon is observed due to the

fact that, over 15 years the stock market goes through large amounts of changes due to external events(Chen et al. (2015)). In order to understand the approach of this research, a smaller time frame is required in order to evaluate the proposition. By considering a time range of over 7 years, a Bidirectional LSTM model has been applied to the American stock market, which yields a result of over 78% with normalised RMSE values in a substantial range(Althelaya et al. (2018)). Furthermore, a shorter time period of around 2 years is used and applied to an ensemble LSTM model. This yields a good result as compared to the previous two models, with an accuracy of over 82%. Apart from the results, an additional layer for outlier identification has also been used and an improvement is seen compared to the results before the identification ensemble layer(Borovkova and Tsiamas (2019)).

2.1.2 GRU - Gated recurrent units

Gated recurrent units are an improvement over the traditional LSTM models due to its capability to process historic data. The result of this feature is that when the dataset is small in number, it increases the probability of re-processing the historic data and thereby increasing the performance of the overall model(Dey and Salem (2017)). One of the applications is observed in the American Stock Market by predicting values for the time period of 2016-2017 and has an accuracy of over 91% (Minh et al. (2018)). By taking advantage of the volume data available in the stock market, which the amount of units bought and sold in a day this method of implementation makes use of multiple features from the data to make up for the absence of data points available. After using this approach, an overall accuracy of over 88% is observed with a lower rate of error(Shen et al. (2018)).One of the conclusions drawn from this review is that the models perform well when outliers values have been eliminated.

2.1.3 CNN - Convolutional Neural Networks

CNN have high performance when its come to predicting values for a dataset which has fewer data points to train, one such application is observed over the Indian Stock Market, where an accuracy of over 86% is observed over a period of 2 years of stock market data. Although the accuracy of this network is high, the amount of time taken to execute 1 epoch is very large(Castoe (2020)).To overcome the training time required to train a Convolutional Neural Network, a new version of CNN has been tested called as Echo state Network.Echo State networks(A type of Convolutional neural network) were tested on google's stock and compared to the Kalman filter which gave an interesting result stating that Echo state Networks perform better than and average Kalman filter by 27% (Bernal et al. (2012)). The multi layer version of CNN deployed over a dataset which expands over a period of 1 year performs really well along with a layer to eliminate outliers from the data (Ponnam et al. (2016)). From all the articles explored, it is observed that the performance increases dramatically when outlier detection layer is deployed before applying the ML models to the datasets.

2.2 Machine Learning in Call/Put Options Pricing:

There have been multiple attempts at pricing Call/Put options by using machine learning algorithms. One such approach is to treat the call and out options similar to a stock

price and apply ML models like General Regression Neural Network (GRNN) and Support Vector Regression (SVR). Out of the two models, GRNN has a higher accuracy as compared to SVR. Although these two algorithms are efficient but this study treats the Options as a stock which does not take into consideration the underlying stock(Phani et al. (2011)). On the other hand, machine learning algorithms have been compared to traditional options pricing techniques like the Black–Scholes equation. ML models only marginally perform better than traditional options pricing techniques, when combined with the Black–Scholes model the overall output of the prediction model is improved. The best accuracy obtained in this case is after pairing the Black–Scholes with an ensemble model(Chowdhury et al. (2020)).The approach of combining the traditional technique along with machine learning models is an ingenious approach but does not consider the price movements of the underlying assets as discussed for the previous research. On the lines of the problems statement of this research, consideration of the underlying stock while pricing the options is a critical aspect as it will ensure the exclusion of momentary impulsive movements(Yoshida (2003)).The combination of the price of the underlying asset and the price of options pricing is an efficient way in deciding about the investor sentiment but is not explored deeply in creating profit(Washimi (2020)). After considering above literature it can be considered that the application of machine learning models to determine the price of the underlying asset of a Call/Put option along with pricing that Call/Put option is not extensively explored.

Summary:The reviewed literature helps in understanding two major aspects of this research and provides a clearer picture on the steps that are proposed in the following sections. The primary aspect is the time period that is taken into consideration while training and testing a Machine learning model. As per the available literature the accuracy of the model starts to reduce as the number of days used for prediction are reducing(This implies that there are less data points to test and train the model) and the time taken to train the model is increased significantly. As price prediction in stock market is a highly time constraint due to its stochastic nature, it becomes significantly important to **reduce the training time** of the models as well **increase accuracy** of the results and at the same time predict the price for the next 30 days. Secondary aspect of this research is the application in the domain of financial derivatives and it is observed from the academic literature that this is not extensively explored as ML models have not been applied to the underlying asset and then a decision of investing in a Call Option/Put Option is made(This is explained in briefly in the sections below).

3 Methodology

This section will discuss about the procedures that are followed while selecting and processing data along with an explanation of how the ML models are being applied to this data. Additionally, this section also contains a brief about financial derivative to assist in understanding the applications of this research.

3.1 Exploratory Data Analysis:

An individual stock is susceptible to external factors which influence the movements of the stock.To minimize the effects of such activities, this research takes into consideration a market index.Market index is a weighted sum of top value stocks listed in a particular

market. In the case of this research, NIFTY 50 is one such index and it is a benchmark index of the India stock Market. It represents a weighted average of 50 of the Largest Indian companies which are listed on the National Stock Exchange(NSE). It can be understood mathematically by the below formula:

$$NIFTY50 = \frac{\sum_{n=1}^{n=50}(StockPrice)_n}{50}$$

Here, NIFTY50 is the current value of the index and Stock Price is the price of individuals stock under Nifty 50.

Figure 1: NIFTY 50 Index Value(01-01-2011 to 30-11-2021)

Initially, the period of observation of the NIFTY50 index is select to be from 01-01-2011 to 30-11-2021. The primary reason for choosing this time period is that it encompasses all the events that occurred in the previous decade. Secondly, all the analysis performed on the index value of this time period can be extrapolated to the future movements.**It can be observed from the above figure that the there is an impulsive drop around March of 2021 which is due to Covid-19**. To avoid such impulsive flash moves in the market price, it is of utmost importance to detect outliers and to process the data accordingly.

3.2 Data pre-processing

In order to get an accurate and efficient machine learning model, it is important to identify and eliminate Outliers in the data. As the number of features available in a stock market dataset is less is number it would be ideal to consider a technique which would yield positive results while working with less amounts of features. In a broader perspective a outlier detection problem is a classification problem as a data point can be classified either in one or the other category(Marghny and Taloba (2014)).In addition

to the less number of features that are available, the number of data points available as relatively small in number. In such a scenario, a good method to detect outliers is the K-means Clustering and K-median method(Angelin and Geetha (2020)). In order to identify Outliers in the NIFTY50 values dataset, the total time range of 2011-01-01 to 2021-11-30 is considered.

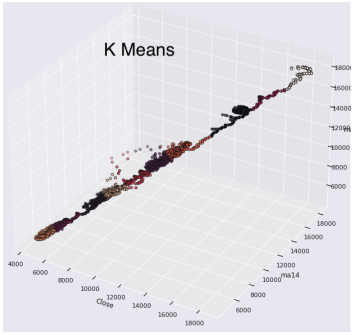


Figure 2: K-Means 3D(Outlier Detection)

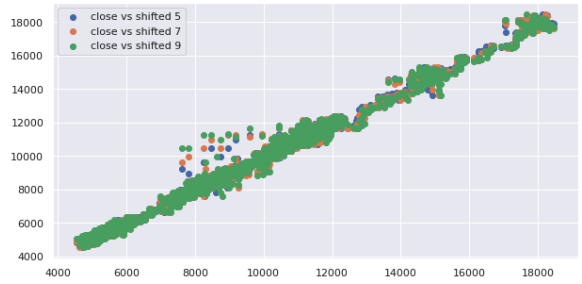


Figure 3: Outliers Range - Compilation

In the above figure, it is evident that values that have been impulsively moved to a range between 7,000 and 10,500 have been identified as Outliers from the period of 2011-01-01 to 2021-11-30. This can also be confirmed by plotting these values versus the values which are shifted by 5,7 and 9 days in order to get a rolling mean. These points are plotted on the time series in order to get a clearer picture as shown below. The red dots showcase the relative outliers in the time series which can lead to inefficiencies in the model.

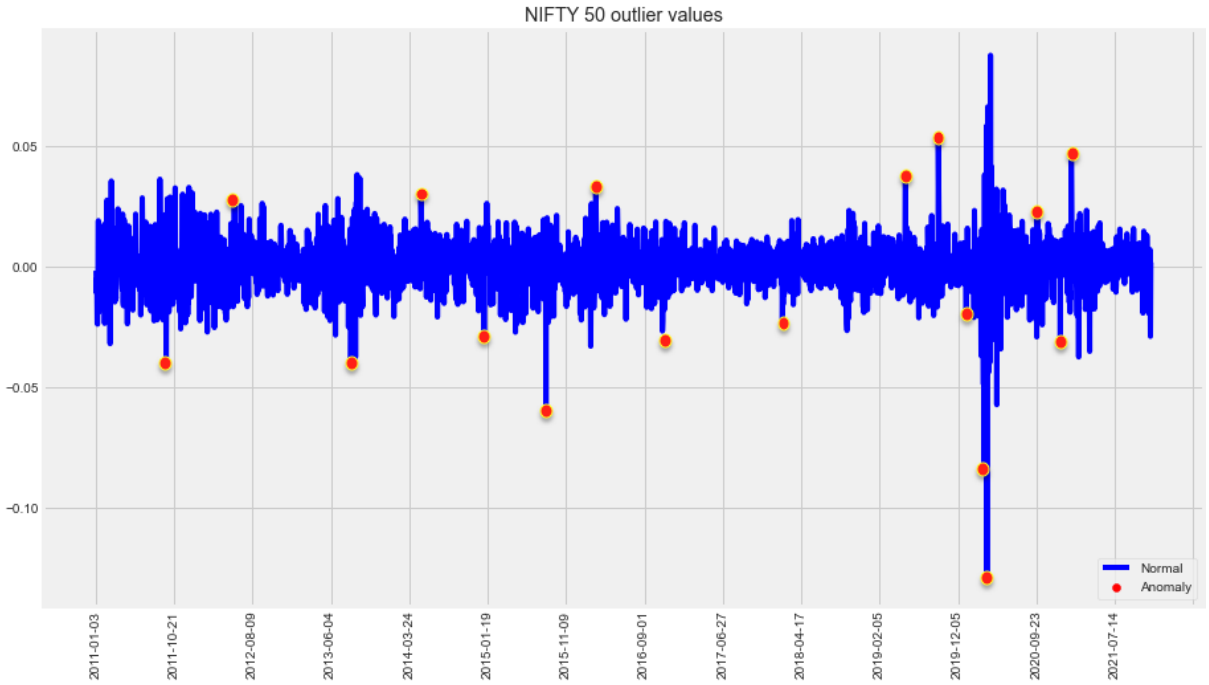


Figure 4: NIFTY 50 Index Value - Outliers

After examining these values, it can be deduced that there are two separate periods of times which can be sampled for this research as it meets the research criteria in the

number of days and absence of extensive outliers. *These two separate time periods are (01-01-2016 to 12-31-2017) and (01-01-2021 to 11-30-2021). The first time period is of exactly two years and the second time period is for 11 months and will be used to evaluate the models of this research.*

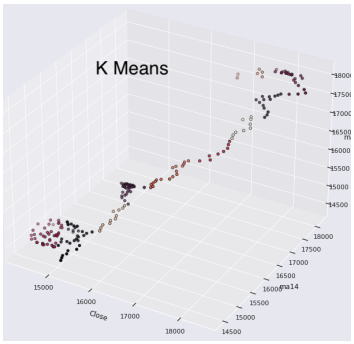


Figure 5: K-Means 3D(Outlier Detection)

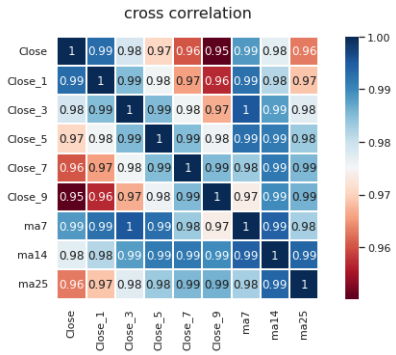


Figure 6: Correlation Matrix - Close Value

Similar methods of identifying outliers have been applied to these two sets of data in order to get a clearer picture on these two separate time periods. The reults of these models are as shown in the figures above. In order to adhere to the latest market trends and to make sure that this study is relevant in the year 2021, time period ranging from 1st January 2021 to 30th November 2021 is selected.

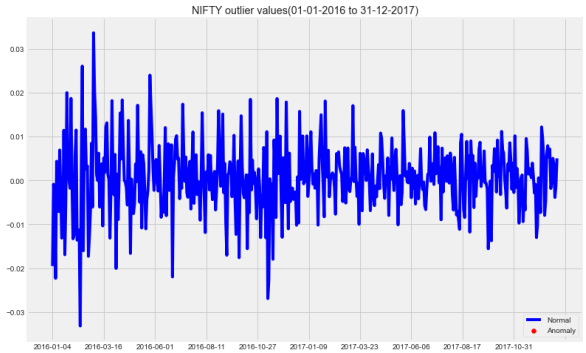


Figure 7: (01-01-2016 to 31-12-2017)

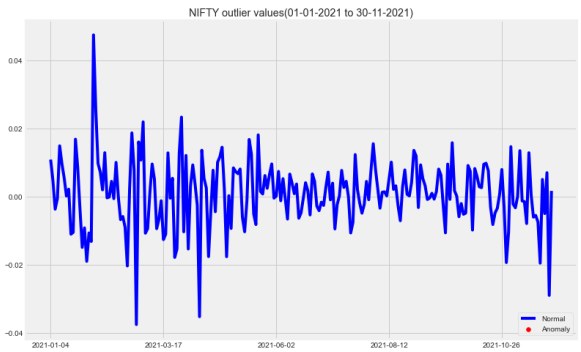


Figure 8: (01-01-2021 to 30-11-2021)

As seen that there are no significant outliers in this period of time which will ensure that the models predict the price with higher accuracy. ***The results of this research will be evaluated for the time period of 01st December 2021 to 31st December 2021 (30 days)***.

3.3 Model Application Methodology:

LSTM - Long Short Term Memory: is a specialised form of a Recurrent Neural Network which is used because of its ability to eliminate the vanishing gradient problem while handling long-term dependencies. In order to resolve this issue of RNN, a cell state c_t is maintained which keeps track of the current state of the model as well as maintains information for a long sequence of time.

$$i_t = \sigma(x_t U^i + h_{t-1} W^i) \qquad f_t = \sigma(x_t U^f + h_{t-1} W^f) \qquad o_t = \sigma(x_t U^o + h_{t-1} W^o)$$

$$g_t = \sigma(x_t U^g + h_{t-1} W^g) \qquad h_t = tanh(c_t.o) \qquad c_t = c_{t-1}.f + g.i$$

To calculate the cell state, two gates are used namely 'f_t' and 'g_t' which are multiplied by the 'i_t' input gate.

GRU - Gated Recurrent Unit: is considered to be a modified version the LSTM model which has two gates out of which one is the rest gate 'r_t' along with an update gate 'z_t'. These two gates are formed in order to determine how the current input in the model is combined with the historic input of the model. In this case the update gate is responsible for the amount of historic data that enters the model in order to help the model to learn. Backpropagation technique in time assists in training the weights of the rest gate and the update gate given by:

$$r_t = \sigma(x_t U^r + h_{t-1} W^r) \qquad z_t = \sigma(x_t U^z + h_{t-1} W^z)$$
$$k = tanh(x_t U^k + (h_{t-1}.r) W^r) \qquad h_t = (1 - z).k + z.h_{t-1}$$

In case of this research, a multilayered model will be developed which will run for three different epoch conditions and have a test size of 30 data points. As per this requirement the above equations will be evaluated for a range of *t+30* and RMSE along with accuracy will used to evaluate the models.

CNN - Convolutional Neural Network: is an improvement on the previously mentioned models and the convolutional layer is added to perform convolutional operations on the data. In this case the input is considered as a function and another function is applied as a filter in order to measure the convolution and apply variations.

$$V_{i,j}^l = \delta(\sum_{k=0}^{F-1} \sum_{m=0}^{F-1} W_{k,m} V_{i+k,j+m}^{l-1}) \qquad f(x) = max(0, x)$$

The input of layer l is calculated as per the above formula and a function is applied to obtain the value of $v_{1,1}$ in the next layer. Apart from this the output of each layer is passed through an activation function.

3.4 Brief About Call/Put Options(Financial Derivatives):

To understand the Call/Put options, it is initially important to understand the entire picture of the financial derivatives domain. A financial derivative is an entity whose price changes when the price of the underlying asset changes(Hirsa and Neftci (2013)). There are are four major types of Financial derivatives namely Forward,Futures,Options and Swaps as shown in the figure below.As per the requirement of the this research, "Options" are the only type of financial derivatives that are under observation. Under Options, there are two types namely "Call Options" and "Put Options" as shown in the figure.

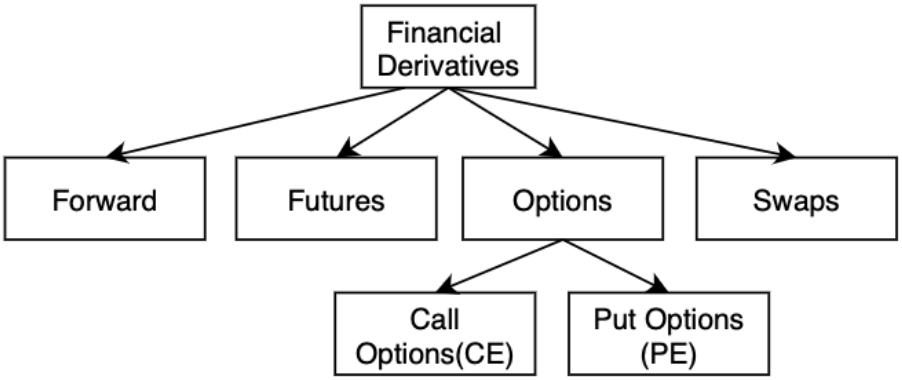


Figure 9: Classification of Financial Derivatives

To understand these two entities, Consider the Google Stock price to be the underlying asset and as the price of the Google Stock changes, so does the price of the financial derivative.As seen in the figure below, the Call Option(CE) is directly proportional to the price of the stock and as the price of the Stock increases the Value of the Call Option also increases.

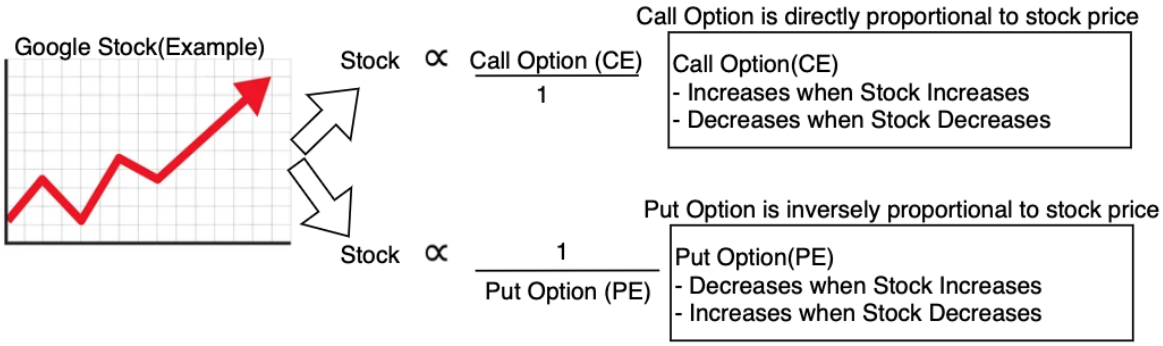


Figure 10: Proportionality Relationship of CE/PE with Stock Price

Similarly, as the price of the Stock decreases the price of the Call Option also decreases. On the other hand, a Put Option(PE) is inversely proportional to the stock price. As the price of the stock increases, the price of the Put Option decreases. Similarly, as the price of the stock decreases the price of Put Option increases.**Due to the availability of this financial derivative, traders and investors can make a profit even if the stock price increasing or decreasing.**

4 Design Specification

This section discusses the techniques that have been deployed in this research in order to improve the results. Along with the techniques, this section also gives an overview of the model architecture and the configuration of program written in order to accomplish this task.

4.1 Parameter Tuning and model Complexity:

Unlike highly complex datasets, a financial market dataset comparatively contains less numbers of features. In order to ensure that the performance of the models and its applications over a wide range of time frames is not compromised, parameter tuning is performed. In the preliminary training and testing of the model, it was discovered that the performance of the model starts to increase as the number of epochs increases from 1 to 10-12 but then it starts to reduce dramatically(Paziewski and Wielgosz (2014)). In order to get maximum performance from the environments and the model, the **models are tested in the range of 1-10 Epochs**.

4.2 Input normalisation and regularisation:

In order to deal with geometric biases and to distribute the importance of values the input is normalised. Additionally, normalisation is achieved in order to ensure that all values are situated in the similar range in order to make them comparable and achieve a highly efficient model. The training dataset is normalised using a element wise min-max scaling of the data with the help of below formula.

$$X_{norm} = \frac{X - Xmin}{Xmax - Xmin}$$

Another issue that is being dealt with here is the issue of overfitting by taking regularization approach of the data, i.e due to the unneeded idiosyncrasies of the training data there is poor generalisation. Along with l_2 regularisation and early stopping, overfitting and unnecessary complexity is prevented. On the other hand momentum is applied in order to avoid stochastic gradient descent termination in local minimas with small spaces.

$$rate^* = rate.\frac{1}{1 + decay.epoch}$$

The above formula is used to utilise dynamic learning rate decay in order to find a minimum along the descent path of the optimiser.

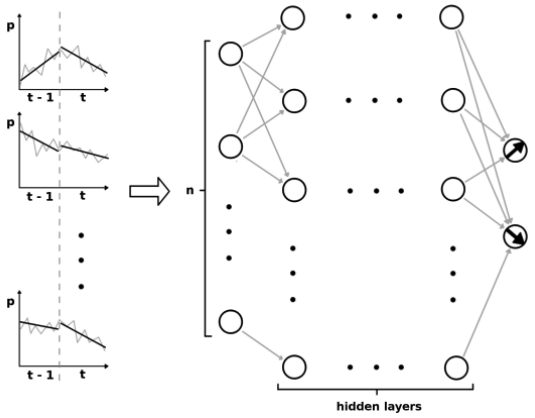


Figure 11: Model Complexity

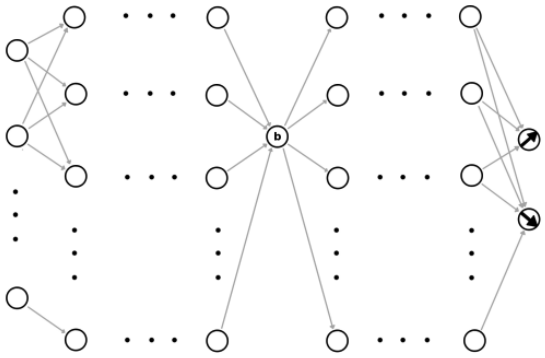


Figure 12: Bottleneck Reduction

4.3 Reduction of Complexities via Bottleneck Layers:

In the context of this research, the complexity increases with the reduction of features as there is an increased chances of bottlenecks appearing in the data model. In order to deal with bottleneck, an extra layers is introduced in the model to exclude remnants of outliers which were present in the dataset after data cleaning.This approach is favoured by making use of autoencoders as the preceding step for the model, due to the fact that autoencoders learn a no goal reduction of the inputs given to them (Gehring et al. (2013)).The configuration of the layer is as shown in the above figure, it is present after the preliminary model building and as a pre-penultimate layer in the entire multi-layered structure.

4.4 Simulations performed:

Simulations are performed in order to find out all the possible outcome of a scenario.In finance, Monte Carlo Simulations have gained popularity due to its ability to accurately estimate integrals. Monte Carlo Simulations are a sub-category of computational algorithms which implement a repetition of random sampling to solve which has a probabilistic interpretation (Raychaudhuri (2008)). The primary idea behind a Monte Carlo Simulation is to predict the number of possible outcomes of a given situation over a given period of time(30 days in this case). Due to the randomness in the price movement of the stock market, these simulations are based in a stochastic differential equations (SDE). A Stochastic process follows a motions known as the Geometric Brownian Motion (GBM) given by:

$$dS = \mu S dt + \sigma S dW_t$$

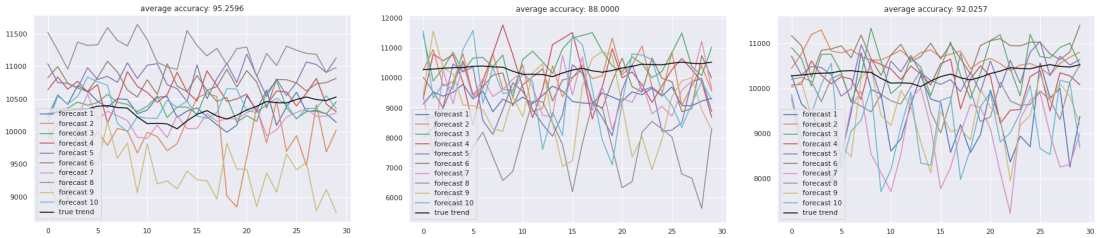
Here, S = The price of the stock/ Value of the index, σ = The Volatility of the stock also known as the diffusion coefficient, μ = Average return over a given period of time along with the instantaneous expected return. Historically, the issue with implementing a Monte Carlo simulation was that it required a large amount of computing power, but this has changed over the years and now Monte Carlo Simulation can be applied in order to get average values of the resulting output which can be further corresponded to the output of the models that are going to be applied in this research(Zio (2013)).**Thus, a simulation will assist in achieving the prediction range of the future price/value for the models implemented in this research. The Deviation range and the Close price range can be fairly judged by the output of a Monte Carlo Simulation.**

5 Implementation

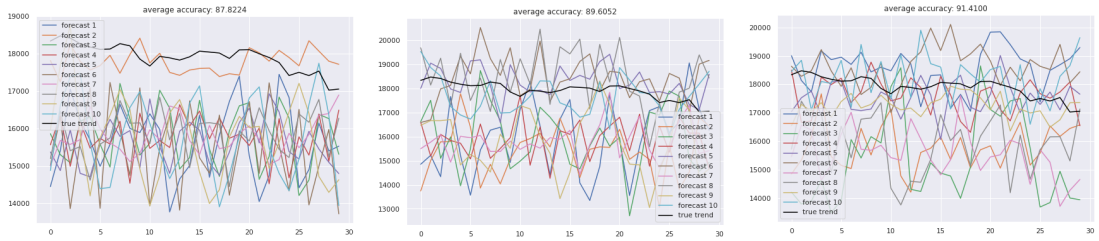
This section discusses the execution and implementation of this research along with some initial application of Baseline Models for NIFTY50 dataset and Simulations which are considered in order to compare the output of this research. In order to judge the output of a ML, it is important that the baselines are measured as per the requirements of the dataset used in this research. Similarly, the simulations are applied to the dateset in order to approximately judge the arc of the output and the range along with the standard deviation in the range.

5.1 Baseline Models:

The baseline models that were available in the published research were implemented for a wide variety of the datasets. In order to taylor the baseline models for this research, the models were applied to the NIFTY50 dataset for the two time periods under observation which are from 2016-2017 and 2021(Till November).The below table summarizes the results of these simulations along with their outputs.



The above figures shows the three models being applied to the time period of 01-01-2016 to 31-12-2017. The figures below shows the three models being applied to the time period of 01-01-2021 to 30-11-2021.



In order for the baselines to be in line with the research question, **all the models are tested to make predictions for the next 30 days.**

Table 1: Results for Vanilla models

Models	Time Period	Epochs	1	5	10	Avg.RMSE
Vanilla - No layers	2016-2017	Acc%	23.56%	63.52%	89.62%	751.17
	2021	Acc%	35.69%	76.52%	86.32%	744.76
Bidirectional Vanilla	2016-2017	Acc%	52.33%	79.54%	95.15%	732.7
	2021	Acc%	73.65%	86.54%	93.67%	562.02
Vanilla-2 path	2016-2017	Acc%	59.68%	64.56%	79.90%	537.23
	2021	Acc%	48.66%	51.64%	76.48%	624.24

The table of observation shows that the overall performance of the bidirectional model is relatively better than the simple model and the 2-path model.

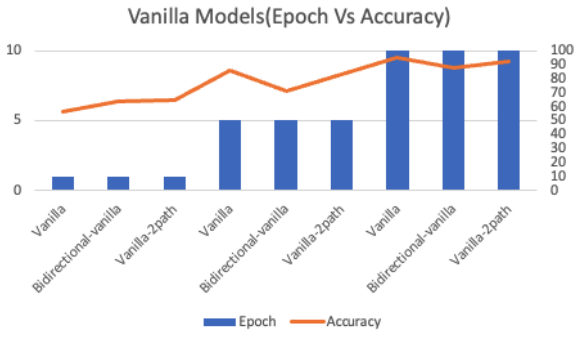


Figure 13: Epoch vs Accuracy

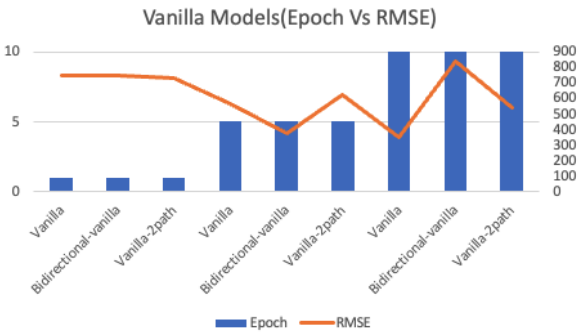


Figure 14: Epoch vs RMSE

As per the above plot, it is evident that the **Bidirectional model type has a good balance between the Accuracy and RMSE values for every epoch range.** The evaluation section applies all the models to the two distinct datasets in order to obtain the model with highest performance.

5.2 Simulations

Simulations are performed initially before implementing the models in this research as the results of these simulations can be harnessed in order *to calculate the percentage of time an event has occurred, along with the average value of the features as required at resulting time period.*

$$d(Value) = \mu Value.dt + \sigma Value.dW_t$$

Here, Value = The Closing Value of NIFTY50, σ = The Volatility of the stock also known as the diffusion coefficient, μ = Average return over a given period of time along with the instantaneous expected return.

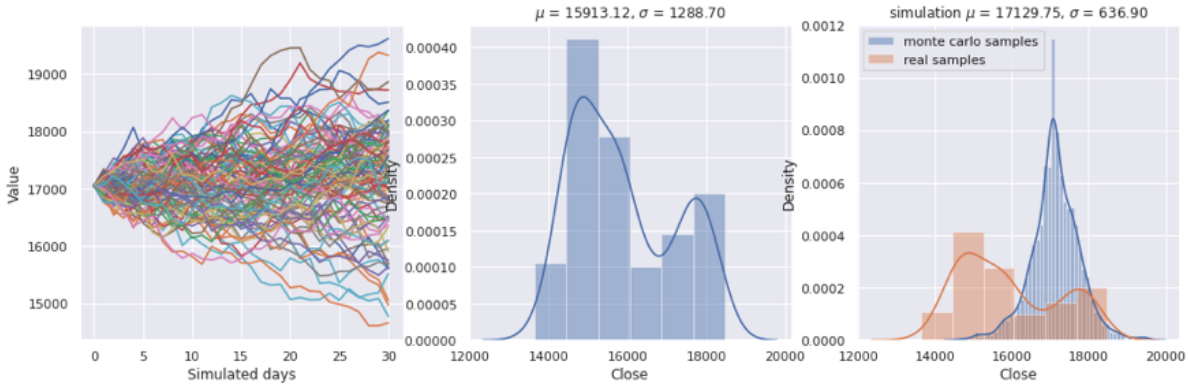


Figure 15: Simple Monte Carlo - No changes to the Close Value

The above diagram applies a simple Monte Carlo Simulation with the input as the dataset of this research. The date range of this simulation is from 01-01-2021 to 30-11-2021 and it simulates for the next 30 days.

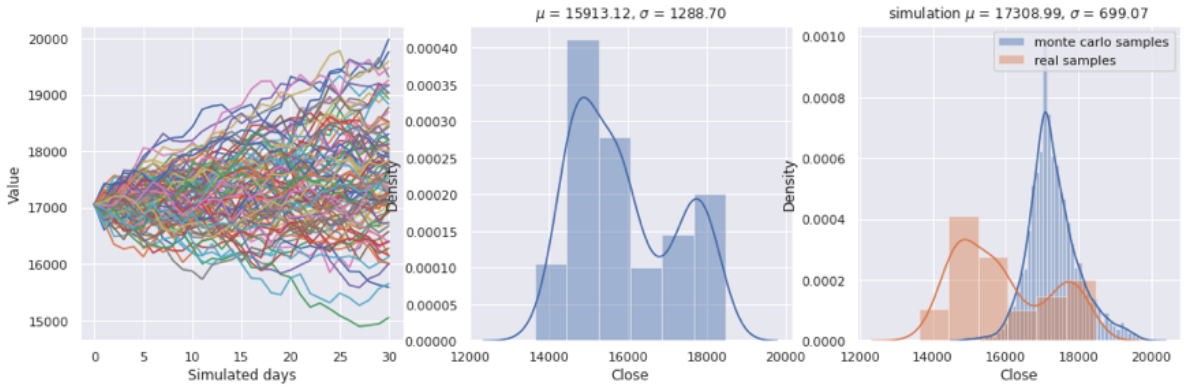


Figure 16: Monte Carlo Drift - Shifted Close prices by K-means

The above diagram applies a Drift Monte Carlo Simulation with the input as the dataset of this research. The date range of this simulation is from 01-01-2021 to 30-11-2021 and it simulates for the next 30 days.

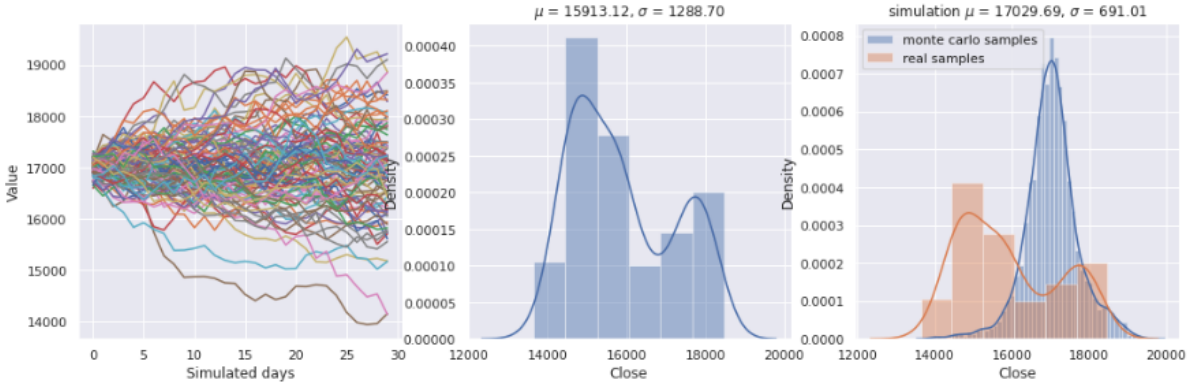


Figure 17: Monte Carlo Dynamic Volatility - Varying Close price changes

The above diagram applies a Dynamic Volatility Monte Carlo Simulation with the input as the dataset of this research(Bonate (2001)). The date range of this simulation is from 01-01-2021 to 30-11-2021 and it simulates for the next 30 days.The results of the above simulations have been summarized in the table below:

Table 2: Monte Carlo Simulation Results:

Simulation	Real σ	Simulated σ
Simple Monte Carlo	1288.70	636.90
Monte Carlo Drift	1288.70	699.07
Monte Carlo Dynamic Volatility	1288.70	691.01

As the Monte Carlo simulation was run on the same dateset, the value of Real σ is the same for all the three simulations. For the application of this research a Simple Monte Carlo simulation would yield efficient results with a value of $\sigma = 636.90$. **All the models in the later sections will be compared to the values of these simulations in order to get the range of error in the model.**

5.3 Approach towards Implementation and Deployment:

The below figure summarizes the approach taken towards the implementation of all the models in this research. The code has been developed and deployed in such a way that all the modules of the code shown below are highly modular in nature. This means that the modules are interchangeable among all the models as the variables and libraries are used with uniformity(Soni et al. (2020)).

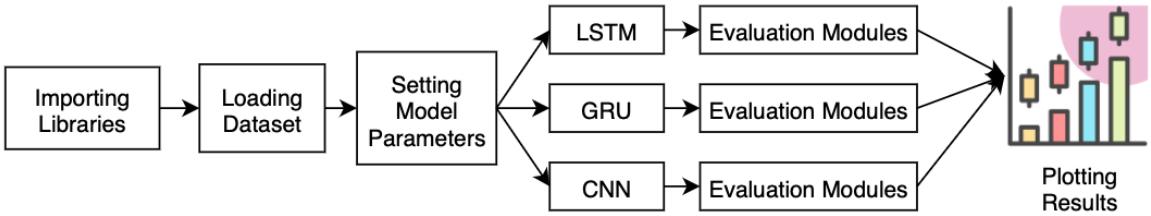


Figure 18: Flow of the implementation

This approach takes places in 6 separate steps in which the first two steps include loading the libraries and the datasets into the Notebook. The next step is to set the parameters for the models as discussed in the previous sections. As this is a common module, the parameters can be set once instead of changing them for each of the models.After the application of each model the evaluation module is common between the models and give vital information about the models such as the Accuracy and RMSE values. At the last stage, these results are plotted in order to get some graphical interpretation of the results.

6 Evaluation and Results

This section gives a comprehensive insight into the results of this research along with the main findings. The evaluation criteria and the methodology used is also discussed in brief in order to provide a clearer picture of the output of the research. Stock market data is predominantly numeric in nature at the same time the nature of the stock market can be considered as a classification problem, in a sense that it either goes up or goes down(Roondiwala et al. (2017)). In addition, to this the machine learning library(scikit-learn) used in this research provides tremendous analytical capabilities to perform analysis on the output(Parmar et al. (2018)). The evaluation schema used in each of this is discussed in brief below in order to get an intuitive insight into the results.

6.1 Evaluation matrix:

The below two methods are used as an evaluation criteria for all the models applied later in this section.

6.1.1 RMSE

In order to evaluate the output of any of the models applied in the research, the value of Root Mean Square is taken into consideration. The ML libraries used in this research provide a precedence for this calculation based on the formula below:

$$MSE = \frac{\sum_{i=1}^{i=n}(O_i - F_i)^2}{n}$$

Here,MSE= Mean Square Error, O_i signifies original closing price, F_i signifies predicted closing price and n = window size. As this gives the Mean Squared Error Value, the Root Mean Squared Error is calculated as shown below:

$$RMSE = \sqrt{MSE}$$

After taking the root of the MSE value, the RMSE value has been calculated. This takes into account the Real and Predicted Closing Values of the stock price/index value it proves to be an efficient way to evaluate a model in this domain(Vijh et al. (2020)).

6.1.2 Accuracy

Accuracy of a model represents the quality and state of being precise in prediction . This is calculated on the basis of the real and predicted values and taking percentages(Pathak and Shetty (2019)). It is given by the below formula:

$$Accuracy\% = (1 - \sqrt{\sum_{i=1}^{i=n} \frac{(R_i - P_i)^2}{(R_i)^2}})X100$$

Here, R_i signifies original closing price, P_i signifies predicted closing price and n = window size. This gives the percentage accuracy of the output of the model.

6.2 Case Study 1 : LSTM - Long Short Term Memory

The LSTM models along with all the variations have been applied to two separate time periods which are (01-01-2016 to 12-31-2017) and (01-01-2021 to 11-30-2021).The below table summarizes the results for all the models under the LSTM domain that have been applied in this research.

Table 3: Results for LSTM models

Models	Time Period	Epochs	1	5	10	Avg.RMSE
LSTM	2016-2017	Acc%	57.6%	74.97%	84.54%	788.21
	2021	Acc%	55.26%	61.58%	78.56%	541.75
BLSTM	2016-2017	Acc%	79.82%	89.66%	95.15%	821.58
	2021	Acc%	76.59%	81.25%	89.67%	529.26
LSTM 2-path	2016-2017	Acc%	51.26%	58.45%	70.90%	651.68
	2021	Acc%	13.52%	48.95%	56.48%	630.47
LSTM Seq-2-Seq	2016-2017	Acc%	75.62%	81.52%	86.94%	480.07
	2021	Acc%	52.63%	58.95%	64.62%	648.26
BLSTM Seq-2-Seq	2016-2017	Acc%	77.23%	86.55%	95.17%	571.98
	2021	Acc%	84.5%	89.65%	93.8%	461.12
BLSTM S2S VAE	2016-2017	Acc%	79.82%	85.96%	92.84%	541.75
	2021	Acc%	66.25%	74.56%	85.90%	478.52

The results of the table are plotted as a graph of RMSE vs Accuracy in order to get an intuitive graphical representation.

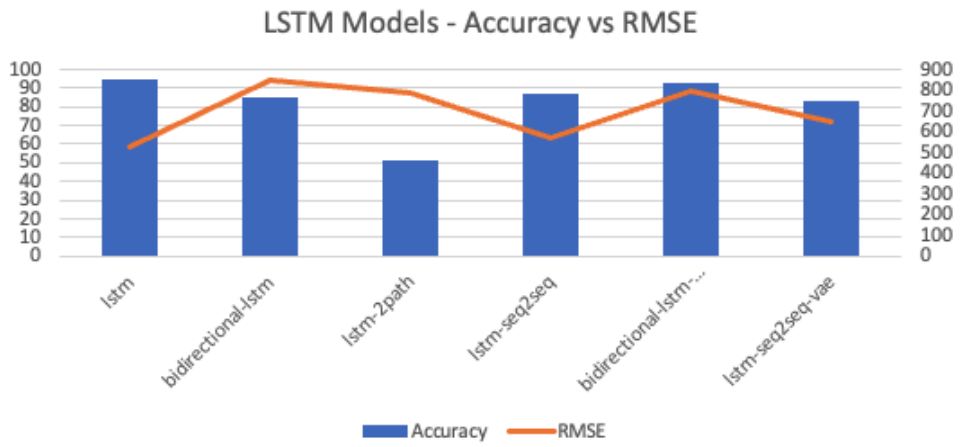


Figure 19: Accuracy vs RMSE for all LSTM models

It can be concluded that the **Bidirectional LSTM and the Bidirectional LSTM Seq2Seq** perform relatively better than the other models in the LSTM domain.

6.3 Case Study 2 : GRU - Gated recurrent unit

The GRU models along with all the variations have been applied to two separate time periods which are (01-01-2016 to 12-31-2017) and (01-01-2021 to 11-30-2021).The below table summarizes the results for all the models under the LSTM domain that have been applied in this research.

Table 4: Results for GRU models

Models	Time Period	Epochs	1	5	10	Avg.RMSE
Simple GRU	2016-2017	Acc%	65.85%	72.55%	89.54%	778.65
	2021	Acc%	43.25%	69.65%	78.56%	492.04
Bidirectional GRU	2016-2017	Acc%	78.95%	88.66%	95.15%	534.17
	2021	Acc%	79.52%	87.64%	91.67%	558.95
GRU 2-path	2016-2017	Acc%	49.56%	68.25%	80.90%	541.91
	2021	Acc%	52.12%	71.56%	86.48%	610.25
GRU Seq-2-Seq	2016-2017	Acc%	38.22%	71.54%	86.94%	550.55
	2021	Acc%	13.52%	26.51%	44.62%	676.65
BGRU Seq-2-Seq	2016-2017	Acc%	84.55%	88.59%	94.70%	451.22
	2021	Acc%	81.58%	86.69%	93.8%	326.59
GRU S2S VAE	2016-2017	Acc%	68.25%	84.55%	92.84%	620.43
	2021	Acc%	65.26%	71.58%	85.90%	809.88

The results of the table are plotted as a graph of RMSE vs Accuracy in order to get an intuitive graphical representation.

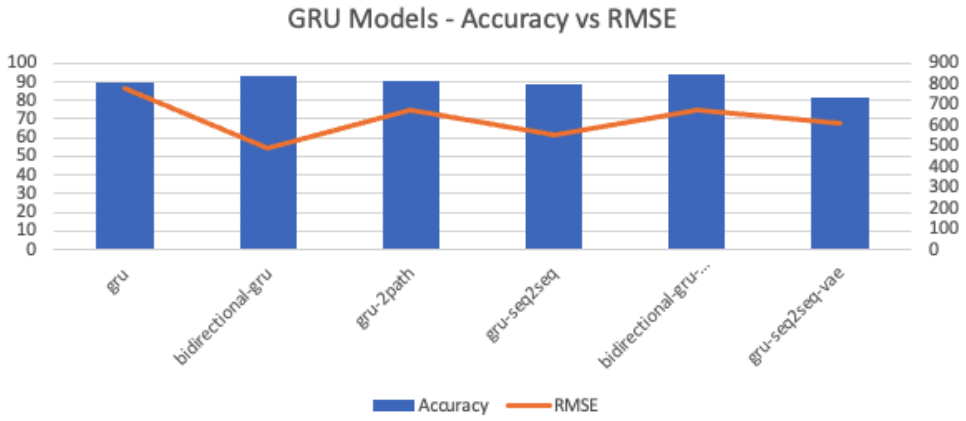


Figure 20: Accuracy vs RMSE for all GRU models

It can be concluded that the **Bidirectional GRU and the Bidirectional GRU Seq2Seq** perform relatively better than the other models in the GRU domain.

6.4 Case Study 3 : CNN - Convolutional Neural Network

The CNN models along with all the variations have been applied to two separate time periods which are (01-01-2016 to 12-31-2017) and (01-01-2021 to 11-30-2021).The below table summarizes the results for all the models under the LSTM domain that have been applied in this research.

Table 5: Results for CNN models

Models	Time Period	Epochs	1	5	10	Avg.RMSE
CNN attention layers	2016-2017	Acc%	71.65%	84.22%	94.54%	417.50
	2021	Acc%	58.56%	79.66%	92.56%	697.82
CNN seq2seq	2016-2017	Acc%	65.23%	86.95%	95.15%	442.8
	2021	Acc%	65.66%	73.56%	89.67%	561.58
Dilated CNN seq2seq	2016-2017	Acc%	41.52%	55.62%	70.90%	616.87
	2021	Acc%	58.25%	69.52%	76.48%	494.54

The table of observation shows that the overall performance of the CNN-Seq2Seq model is relatively better than the other models. In order to further evaluate this, a comparison is made between the RMSE value and the Accuracy for the range of epochs in order to get a clearer picture.

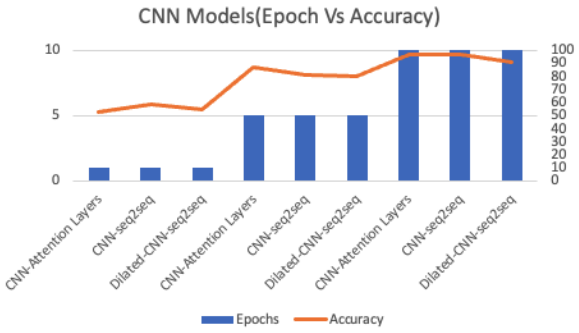


Figure 21: Epoch vs Accuracy

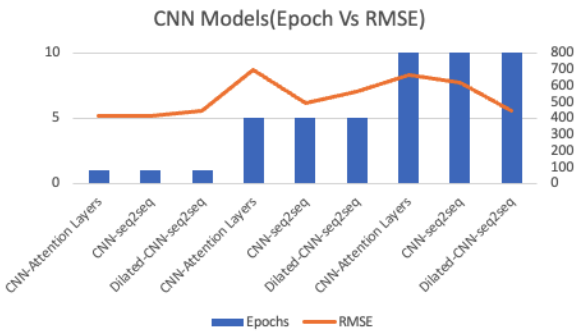


Figure 22: Epoch vs RMSE

As per the above plot, it is evident that the **CNN-Seq2Seq model type has a good balance between the Accuracy and RMSE values for every epoch range.** The evaluation section applies all the models to the two distinct datasets in order to obtain the model with highest performance.

6.5 Discussion and Application

The above experiments performed have achieved results which are in line with the requirements this research. A total of 18 models were tested for this research and were implemented on two separate time frames which were not correlating in nature. This was done to ensure that the performance of the models maintained fidelity throughout. The primary aim of this study was to identify a machine learning technique which would be efficient in predicting the value of a stock or an index over the next 30 days. At the time of this research, the stock market was relative stable after a major crash in March - 2020.

There is one drawback of this research which can be worked upon, it is the major movements in the market which take place due to events such as the Covid-19 or a War in a country. There is academic literature available in order to resolve these issues in ML model and compensate for sudden movements. Apart from this, the current work done is sufficient in terms of the application it is supposed to be applied to in the domain of Financial derivatives.

Application: In order to dive deeper into the results that were achieved in the previous section, the model with the best performance **(BLSTM Seq-2-Seq is considered as it requires the least amount of time to train)** is **applied to the Stock Market Index to predict the movements for the month of December - 2021**. In addition to this,**Monte Carlo Simulation has been run on the output of the model in order to get the range and spread of the predicted output.** This is done in order to demonstrate that the models are valid in the current market and can be used to predict future movements with ease.

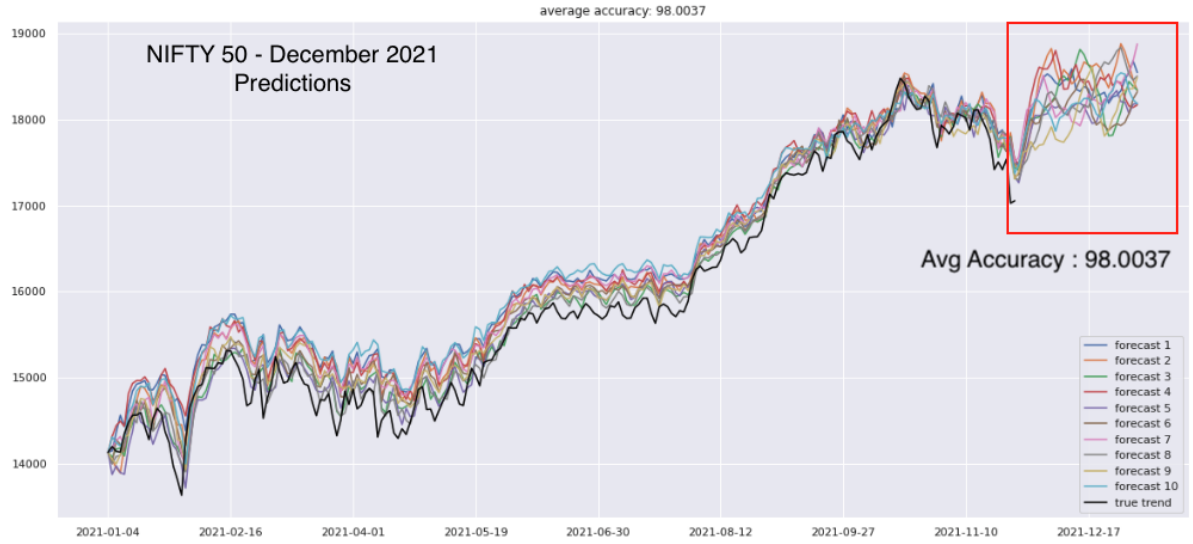


Figure 23: NIFTY 50 Forecast - December 2021

The reason for choosing a time period of 30 days was to identify profitable options(A type of financial derivative). As discussed in Section 3.4, as the price of the underlying asset increases(The above stock index in this case) the price of the "Call Option-CE" will increase as well. From an investors/traders perspective, if the movement of the stock is enough information in order to turn a profit in the market. In the above case, the value of the index was close to *17,000* and the model implemented shows that by the end of december the value will increase above *17,000*. As per these results, the traders/investors can now make a clear trade that states that the market will go in the upward direction.

7 Conclusion and Future Work

The initial proposition of this research states to find whether Machine Learning Algorithms predict the short term movements of the stock market as efficiently as the Long term movements. As per the observed evaluation and results, **it can be concluded that Bidirectional Models along with LSTM is relatively superior amongst other LSTM models in predicting the price for the next 30 days**. Similarly, the **combination of Bidirectional Seq2Seq along with GRU is relatively superior amongst other GRU models**. Additionally it is observed that that the combination of Seq2Seq along with CNN is relatively superior amongst other CNN models but the time required to train the data in CNN models is significantly higher than GRU and LSTM models .Apart from the results of the applied ML models, **the BLSTM Seq-2-Seq model performs quite well in the real world** application to predict an efficient Call Put/Put option in order to maximise a traders/investors profit by a close to 23.2% margin.

Future work related to this research can be conducted in multiple domains like Machine Learning,Finance and Automation systems for traders. In the ML domain, future work can be carried out in order to achieve higher performance models as well as include conditions which cause impulse movements in the market. Similarly in the domain of finance, experts can study the weighted impacts of the Call/Put options at an instance of the market and develop models which include such Biases. On a real world application

front, a complete software can be created which will assist the traders in making decision in the stock market by giving them recommendations. **An example of such an application has been created in order to demonstrate the results of this project as shown below:**

Figure 24: Dashboard Demonstration

This is explained further in the configuration manual as a part of the demonstration.

8 Acknowledgement

I have received a tremendous amount of support and feedback from my thesis Supervisor **Dr. Bharathi Chakravarthi**. He has helped me with his expertise in the domain of Machine Learning and Artificial Intelligence by assisting with my thesis project. Along with my thesis supervisor, I would like to thank the Staff of The National College Of Ireland for their utmost support and coordination.

References

Althelaya, K. A., El-Alfy, E.-S. M. and Mohammed, S. (2018). Evaluation of bidirectional lstm for short-and long-term stock market prediction, *2018 9th international conference on information and communication systems (ICICS)*, IEEE, pp. 151–156.

Angelin, B. and Geetha, A. (2020). Outlier detection using clustering techniques–k-means and k-median, *2020 4th International Conference on Intelligent Computing and Control Systems (ICICCS)*, IEEE, pp. 373–378.

Bernal, A., Fok, S. and Pidaparthi, R. (2012). Financial market time series prediction with recurrent neural networks, *State College: Citeseer* .

Bonate, P. L. (2001). A brief introduction to monte carlo simulation, *Clinical pharmacokinetics* **40**(1): 15–22.

Borovkova, S. and Tsiamas, I. (2019). An ensemble of lstm neural networks for high-frequency stock market classification, *Journal of Forecasting* **38**(6): 600–619.

Castoe, M. (2020). Predicting stock market price direction with uncertainty using quantile regression forest.

Chen, K., Zhou, Y. and Dai, F. (2015). A lstm-based method for stock returns prediction: A case study of china stock market, *2015 IEEE international conference on big data (big data)*, IEEE, pp. 2823–2824.

Chowdhury, R., Mahdy, M., Alam, T. N., Al Quaderi, G. D. and Rahman, M. A. (2020). Predicting the stock price of frontier markets using machine learning and modified black–scholes option pricing model, *Physica A: Statistical Mechanics and its Applications* **555**: 124444.

De Roon, F., Veld, C. et al. (1996). Put-call parities and the value of early exercise for put options on a performance index, *Journal of Futures Markets* **16**(1): 71–80.

Dey, R. and Salem, F. M. (2017). Gate-variants of gated recurrent unit (gru) neural networks, *2017 IEEE 60th international midwest symposium on circuits and systems (MWSCAS)*, IEEE, pp. 1597–1600.

Gehring, J., Miao, Y., Metze, F. and Waibel, A. (2013). Extracting deep bottleneck features using stacked auto-encoders, *2013 IEEE international conference on acoustics, speech and signal processing*, IEEE, pp. 3377–3381.

Greenwood, J. and Smith, B. D. (1997). Financial markets in development, and the development of financial markets, *Journal of Economic dynamics and control* **21**(1): 145–181.

Hirsa, A. and Neftci, S. N. (2013). *An introduction to the mathematics of financial derivatives*, Academic press.

Marghny, M. and Taloba, A. I. (2014). Outlier detection using improved genetic k-means, *arXiv preprint arXiv:1402.6859* .

Minh, D. L., Sadeghi-Niaraki, A., Huy, H. D., Min, K. and Moon, H. (2018). Deep learning approach for short-term stock trends prediction based on two-stream gated recurrent unit network, *Ieee Access* **6**: 55392–55404.

Parmar, I., Agarwal, N., Saxena, S., Arora, R., Gupta, S., Dhiman, H. and Chouhan, L. (2018). Stock market prediction using machine learning, *2018 First International Conference on Secure Cyber Computing and Communication (ICSCCC)*, IEEE, pp. 574–576.

Pathak, A. and Shetty, N. P. (2019). Indian stock market prediction using machine learning and sentiment analysis, *Computational Intelligence in Data Mining*, Springer, pp. 595–603.

Paziewski, J. and Wielgosz, P. (2014). Assessment of gps+ galileo and multi-frequency galileo single-epoch precise positioning with network corrections, *Gps Solutions* **18**(4): 571–579.

Phani, B., Chandra, B. and Raghav, V. (2011). Quest for efficient option pricing prediction model using machine learning techniques, *The 2011 International Joint Conference on Neural Networks*, IEEE, pp. 654–657.

Ponnam, L. T., Rao, V. S., Srinivas, K. and Raavi, V. (2016). A comparative study on techniques used for prediction of stock market, *2016 International Conference on Automatic Control and Dynamic Optimization Techniques (ICDOT)*, IEEE, pp. 1–6.

Raychaudhuri, S. (2008). Introduction to monte carlo simulation, *2008 Winter simulation conference*, IEEE, pp. 91–100.

Roondiwala, M., Patel, H. and Varma, S. (2017). Predicting stock prices using lstm, *International Journal of Science and Research (IJSR)* **6**(4): 1754–1756.

Shen, G., Tan, Q., Zhang, H., Zeng, P. and Xu, J. (2018). Deep learning with gated recurrent unit networks for financial sequence predictions, *Procedia computer science* **131**: 895–903.

Soni, N., Sharma, E. K., Singh, N. and Kapoor, A. (2020). Artificial intelligence in business: From research and innovation to market deployment, *Procedia Computer Science* **167**: 2200–2210.

Vijh, M., Chandola, D., Tikkiwal, V. A. and Kumar, A. (2020). Stock closing price prediction using machine learning techniques, *Procedia Computer Science* **167**: 599–606.

Washimi, K. (2020). Revisiting determinants of investor sentiment in the fx option market by machine learning approaches, *2020 IEEE Symposium Series on Computational Intelligence (SSCI)*, IEEE, pp. 22–27.

Yoshida, Y. (2003). The valuation of european options in uncertain environment, *European Journal of Operational Research* **145**(1): 221–229.

Zio, E. (2013). Monte carlo simulation: The method, *The Monte Carlo simulation method for system reliability and risk analysis*, Springer, pp. 19–58.