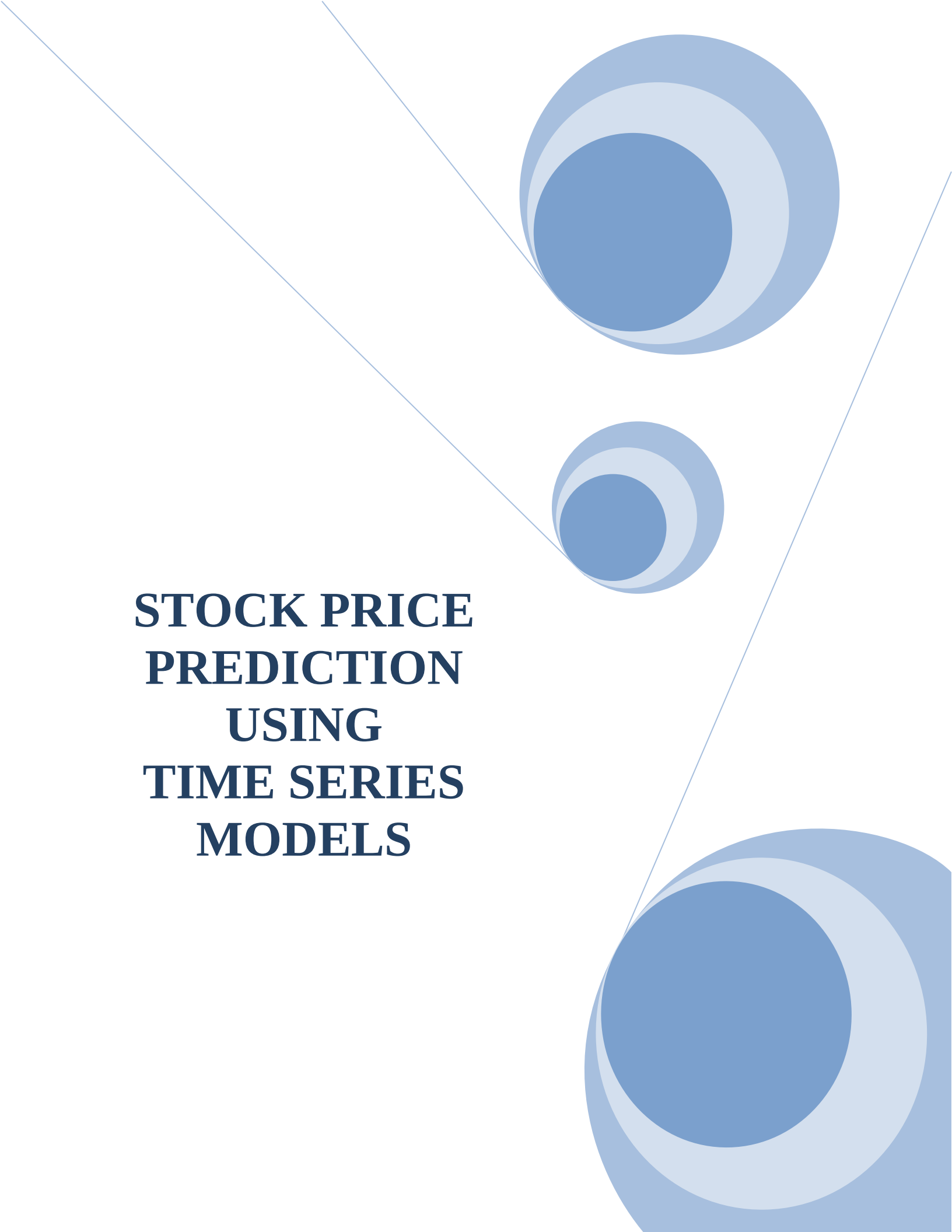


STOCK PRICE PREDICTION USING TIME SERIES MODELS

STOCK PRICE PREDICTION USING TIME SERIES MODELS

Dissertation submitted in partial fulfillment of the requirements

for the degree of

MSc Data Analytics

At Dublin Business School

Shruti Chouksey

Supervisor: Ms. Terri Hoare

MSc Data Analytics 2018 - 19

DECLARATION

I, Shruti Chouksey, declare that this research is my original work and that it has never been presented to any institution or university for the award of Degree or Diploma. In addition, I have referenced correctly all literature and sources used in this work and this work is fully compliant with the Dublin Business School's academic honesty policy.

ACKNOWLEDGMENTS

I would like to express my special thanks of gratitude to Ms. Terri Hoare, my teacher and research supervisor for her patient guidance from the beginning with constructive and valuable suggestions provided throughout the planning and improvement of my dissertation. She also helped me in doing a lot of Research, and her ability to give her time generously has been particularly valued.

Secondly, I would like to thank my family and friends for their support and encouragement throughout my course.

ABSTRACT

This Thesis titled- “Stock price forecasting using time series models” focused on the comparison of the performance of time series models to predict the stock price for 5 banks. Forecasting and stock price analysis is important in finance and economics. Time series forecasting can be applied on any set of variables that change over time. For stocks or share prices, time series forecasting is common to track the price movement of the security over time. There is considerable past research work available on time series forecasting. In this thesis, a comparative study of time series forecasting using 3 models ARIMA (autoregressive integrated moving average), PROPHET and KERAS with LSTM (Long Short Term Memory) models has been explored. Historical stock price data was obtained from the National Stock Exchange (NSE) and used to build these models for comparative purposes. The results obtained reveal that all 3 models have strong potential for prediction and forecasting on the sourced historical data samples. All of the models performed better on larger data samples with LSTM best able to forecast seasonality.

Table of Contents

List of Tables and Figures

List of Important Abbreviations

- ACF- Autocorrelation function
- PACF- Partial autocorrelation function
- AR - Autoregressive
- ARIMA – Autoregressive Integrated Moving Average
- ARMA- Autoregressive moving average model
- LSTM- Long Short Term Memory
- RMSE- Root Mean Square Error
- AIC- Akaike Information Critera
- ts- Time series
- RNN- Recurrent Neural Networks
- SVM- Support Vector MAchine
- ANN- Artificial Neural Networks
- CNN- Convolutional Neural Network
- CRISP-DM - Cross-Industry Standard Process for Data Mining
- EDA – Exploratory data Analysis
- NSE- National Stock Exchange
- BSE- Bombay Stock Exchange
- NYSE- New York Stock Exchange
- CSV- Comma-separated values
- HTML- Hypertext Markup Language
- PDF- Portable Document Format

1 INTRODUCTION

"Prediction is a very difficult art, especially when it involves the future" -Neils Bohr (Nobel Laureate Physicist).

"Forecasting is the typical process which helps in making statements whose actual outcomes yet to be observed or have not yet been observed" Wikipedia.

1.1 Terms and Definitions

What is time series forecasting?
A time series is a sequence of data points that are listed in time order. Time series analysis is combinations of different methods which help in analyzing time series data so as to get extricate meaningful statistics. Time series forecasting is a methodology that helps the model to predict future values using previously observed values.

Brockwell, P.J., Davis, R.A. and Calder, M.V., 2002, and Granger, C.W.J. and Newbold, P., 2014 stated that time series is a set of observations of N number of elements each of which recorded at a particular time (t). A time series is a sequence of values recorded in order by the time parameter. Such as Economic Forecasting, Sales Forecasting, Stock Market Analysis, Yield Projections, Process and Quality Control, Inventory Studies, Workload Projections, Utility Studies, Census Analysis, medical, meteorology (rainfall, temperature), astronomy and many more. Most of the classical statistical techniques and methods are not relevant in time series studies, so the new techniques Time Series analysis have been devised.

Time series forecasting (Granger, C.W.J. and Newbold, P., 2014) is a technique of drawing inference and making a prediction of the future. These techniques required to set up some initial hypothetical probability models, estimate parameters, check for the goodness of fit to the data and possibly to use the fitted model to enhance the understanding of the mechanism generating the series. Once a satisfactory model has been developed, it may be used in a variety of ways depending on the particular field of application. There are many methods used to model and forecast time series. The application type and preference decide the selection of the appropriate technique.

Basically, (Montgomery, D.C., Jennings, C.L. and Kulahci, M., 2015) there need to distinguish between a forecast and predicted value of y_t that was made at some previous time period, and a fitted value of y_t that has resulted from estimating the parameters in a time series model to historical data. The forecast error that results from a forecast of y_t that was made at time period $t - \tau$ is the lead $-\tau$ forecast error.

$$e_t(\tau) = y_t - \hat{y}_t(t - \tau)$$

For example, the lead -1 forecast error is

$$e_1(1) = y_t - \hat{y}_t(t - 1)$$

The difference between the observation yt and the value obtained by fitting a time series model to the data, or a fitted value $\hat{y}_t$ defined earlier, is called a residual, and is denoted by

$$e_t = y_t - \hat{y}_t$$

What is stock?
A stock (Akhilesh Ganti, Mar 2019) (also known as "shares" or "equity) is a kind of security that implies proportionate proprietorship in the issuing organization. This qualifies the investor for that extent of the company's advantages and profit. Stocks are purchased and sold dominatingly on stock trades, however there can be private deals also, and the establishment of about each portfolio.

These transactions have to conform to government regulations which are meant to protect investors from fraudulent practices. Historically, they have outperformed most other investments over the long run. These investments can be purchased from most online stock brokers.

An analysis (Lo, A.W. and Wang, J., 2001) begins with *I* investors indexed by $i = 1, \dots, I$ and *J* stocks indexed by $j = 1, \dots, J$. There is always an assumption that all the stocks are risky and non-redundant. For each stock *j*, let N_{jt} be its total number of shares outstanding, D_{jt} its dividend, and P_{jt} its ex-dividend price at date *t*.

For notational convenience and without loss of generality, assume throughout that the total number of shares outstanding for each stock is constant over time, i.e. $N_{jt} = N_{j,}\ j = 1, \dots, J$.

For each investor *i*, let S_{jt}^{i} denote the number of shares of stock *j* he holds at date *t*.
Let

$$P_t \equiv [\, P_{1t} \ \dots \ P_{Jt}]T$$

Which denote the vector of stock prices and shares held in a given portfolio, where A_t denotes the transpose of a vector or matrix A.

Finally, denote by V_{jt} the total number of shares of security *j* traded at time *t*, i.e., share volume, hence

$$V_{jt} = \frac{1}{2} \sum_{i=1}^{I} |S_{jt}^{i} - S_{jt-1}^{i}\,|$$

Where the coefficient $\frac{1}{2}$ corrects for the double counting when summing the shares traded over all investors.

Time series analysis for stock price trends:
Time series can be applied on any set of variables that change over time. For stocks or share price, time series is common to track the price of a security over time. This can be tracked over the short term, such as the price of a security from time of open to close of a business day, or closing price of a daily business day or maybe last day of every month over the course of last 15-20 years. The seasonal trend and flow is the highlight of the stock market.

This can be useful to see how a company capital, security or any related economic variable changes over time. It can also be used to examine how the changes associated with the chosen data point compare to shifts in other variables over the same time period.

For example, suppose analyzing a time series of daily closing stock prices for Axis bank stock from National Stock Exchange Index India over a period of ten years. Obtain a list of all the closing prices for the stock from each business day for the past ten years and list them in chronological order. This would be ten years all closing price time series for the stock.

The fig 1.1 shows the graphical presentation of stock price over the period of more than ten years.

Fig. 1.1: Axis Bank stock price from NSE Index

1.2 Dissertation Roadmap

This thesis covered forecasting models, the algorithms used within the model and other optimization techniques used for better performance and accuracy for time series dataset. The various performance evaluation parameters applied for automatic selection of proper input variables and model-dependent variables and optimizing the model parameters simultaneously to compare the results with those of other various methods, which show the effectiveness of the proposed approach for the seasonal time series.

The proposed method is a model-independent approach. Here main focused was on predicting price forecasting for any particular bank using the daily closing price and identified trends and patterns in the data using different algorithms. For this RNN (Recurrent Neural Networks) approach LSTM (Long Short Term Memory) was used for the stock price prediction of NSE (National Stock Exchange) listed banks (HDFC, AXIS, ICICI, KOTAK, and SBI) and compare their performance. Also applied different algorithms- ARIMA model, PROPHET algorithm, and KERAS with LSTM (Long Short Term Memory) model for predicting future values on a short term basis and performance of the models was quantified using RMSE error.

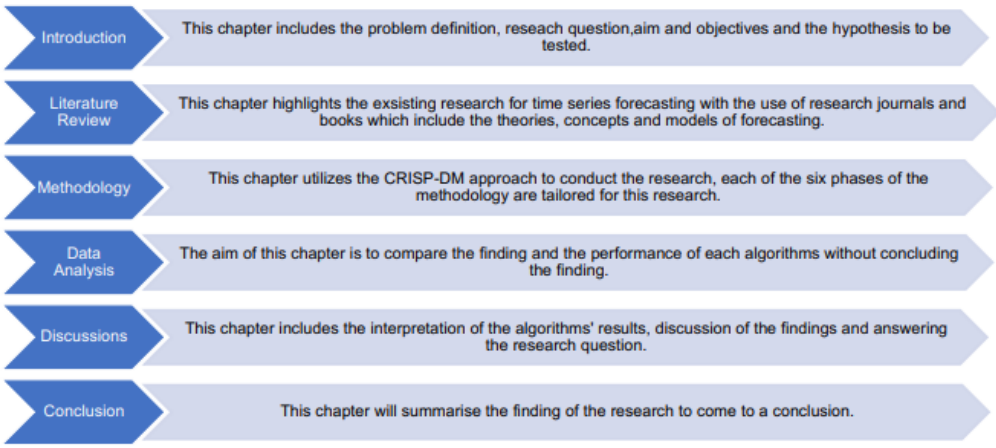


Fig. 1.2: Dissertation roadmap

2 LITERATURE REVIEW

2.1 Business Importance

Stock market or equity market has an overpowering impact in today's economy. A rise or fall in the share price has an important role in determining the investor's gain. Stock market forecasting is always very tricky. No one can see the future– the world is inherently uncertain and surprising things will happen. Even if it is known what's going to happen, however, you might not know how markets will respond.

Investment in the stock market is regarded as high risks and high gains and so attracts a large number of investors and economists. However, information regarding a stock is normally incomplete, complex, uncertain and vague, making it a challenge to predict the future economic performance. People invest in the stock market based on some analysis.

2.2 Scope

Forecasting can be defined as the prediction by analyzing the historical data for many areas including business and industry, economics, environmental science, and finance. Forecasting of time series data provides the organization with useful information that is necessary for making important decisions.

Many of the forecasting's involves the analysis of time. A time series of data for forecasting can be defined as a chronological sequence of observations for a selected variable (stock price). It can either be univariate or multivariate. Univariate data includes information about only one particular stock whereas multivariate data includes stock prices of more than one company for various instances of time.

Also, the analysis of patterns helps in identifying the best-performing companies for a specified period. This makes time series analysis and forecasting an important area of research.

Stock prices are more informative when the information it contains has less social value. Stock market prediction is always uncertain rather an analysis of time series data helps in identifying patterns, trends and periods or cycles existing in the data. It is shown that there is a fundamental tension between the in formativeness of stock prices and the effectiveness of corporate governance, which limits the disciplining role of stock prices.

2.3 Previous Research Analysis

The ARIMA model is one of the most widely used techniques for analyzing time series data points or to predict future data points. One research study of 2004 (Ariyo, A.A., Adewumi, A.O. and Ayo, C.K., 2014) has compared ARIMA models with Support Vector Machine and proposed a hybrid methodology that exploits the unique strength of the ARIMA model and the SVM model in forecasting stock prices problems. They have used real data sets from stock prices to examine the forecasting accuracy of the proposed model.

The hybrid model proposed is composed of the ARIMA component and the SVMs component. Thus, the model can model linear and nonlinear patterns with improved overall forecasting performance. The presented model is believed to greatly improve the prediction performance of the single ARIMA model in forecasting stock prices. This study demonstrated that a simple combination of the two best individual models does not necessarily produce the best results. Therefore, the structured selection of optimal parameters of the model is of great interest.

A comprehensive study of A. Victor Devadoss and T. Antony Alphonnse Ligori, 2013 was done on the common parameters to design a neural network for forecasting economic time series data. The study identifies Artificial Neural Networks are a highly flexible function that can map any nonlinear function and are being used widely in many different fields of business and industry. One of the major application areas of ANN is forecasting.

They implemented nonlinear technique ANN to forecast the stock prices of BSE (Bombay Stock Exchange) stocks and concluded that predicted results of ANN was successful with better accuracies which clearly ensured that ANN is suitable and perform better for forecasting. A clear observation says that in order to improve the performance of the network macroeconomic factors and technical analysis indicators may be used as input variables. They also used Multilayer feedforward network with back propagation algorithm for forecasting purpose which has multiple layers in between and data flows in one direction from input to output layer.

Research paper of Ariyo, A.A., Adewumi, A.O. and Ayo, C.K., 2014 presented the extensive process of designing time series models for stock price prediction using the ARIMA model. They have used the New York Stock Exchange (NYSE) and Nigeria Stock Exchange (NSE) data for research. They used Eviews analytical software and picked up data stock data from two different countries. The data composed of four elements, namely: open price, low price, high price, and close price respectively. For research, the closing price was chosen to represent the price of the index to be predicted. The closing price was chosen because it reflects all the activities of the index in a trading day. Analyzing their results revealed that the ARIMA model has short term potential for prediction. As per this study, ARIMA models can compete reasonably well with emerging forecasting techniques in short-term prediction.

X. Ding, Y. Zhang, T. Liu, and J. Duan, 2015 demonstrated that deep learning method was useful for event-driven stock market prediction by using deep convolution neural network model with the combined influence of long-term events and short-term events on stock price movements, proposing a neural tensor network for learning event embeddings. They focused on automatically learn embeddings for structured events which shows more fundamental relations between events, even if the same object or event is not shared between events. Various studies have found that financial news may have an impact on share price hence they used structured events to represent news, which represents objects of the event. Their experimental result represents that event embedding is useful for the task of stock price prediction. However, the event embeddings based methods give more accurate performance than event-based methods and deep convolution neural network might help in capturing long term influence of news event than standard feedforward neural network.

There are models that are too simple to catch all of the available information. On the other hand there are methods with more parameters employed in order to cope with more demanding underlying patterns; unfortunately, while optimizing all these parameters usually these complex methods end up actually over-fitting the actual data. So, this approach aims to help the models capture the data. This is achieved by breaking the data down into several simpler series, each one of which captures part of the information included in the original series. As a result of this process, simpler models can adapt to these simpler series.

Menon, V.K., Vasireddy, N.C., Jami, S.A., Pedamallu, V.T.N., Sureshkumar, V. and Soman, K.P., 2016 analyzed NSE (National Stock Exchange) stocks using linear models (AR and ARMA). The study identifies that AR predictions are positive whereas ARMA has an enhancement trend. Their prediction worked well for almost all of the stock despite in case of stock splits.

Another Insight they have gained was AR predictions were positive at best while ARMA had an exaggeration trend. So if the prediction is a rise, then the AR value will better quantize that and if it's a fall then it can bank on the ARMA forecast value. So the conclusion is if a prediction is a rise then AR value will be more accurate otherwise can rely on ARMA forecast value if a prediction is a fall.

Hiransha, M., Gopalakrishnan, E.A., Menon, V.K., and Soman, K.P., 2018 experimented stock price prediction using three deep learning models (RNN, LSTM, CNN) and they concluded that the CNN (Convolutional neural network) performs better and gives more accurate results than other two models. The reason is that CNN doesn't depend on any previous information for prediction. CNN helps the model to understand patterns and able to capture trends occurring in the current window since CNN follows the current window for prediction. To compare the error percentage they used linear model (ARIMA) for forecasting. They also observed that changes occurring in the stock market may not follow the same cycle every time and the existence of the trend may be unlike sector wise. Such an analysis of trends and cycles will surely help investors to gain more profit.

LSTM (C Olah, C., 2015) (Long Short Term Memory) is the most popular RNN (Recurrent Neural Networks) approach and they are networks with loops in them, allowing information to persist. These loops make recurrent neural networks seem kind of strange. A recurrent neural network can be thought of as multiple copies of the same network, each passing a message to a successor.

KERAS (Choi, K., Joo, D. and Kim, J., 2017) used deep learning in audio informatics research. For KERAS, preprocessing data frequently involves bunches of time and effort. It was difficult when managing audio data than images or texts because of its vast size and overwhelming interpreting calculation. Hence they utilized a generator. To actualize a generator that loads the data they included a KAPRE layer at the input side of the KERAS model. Their main focus behind using KAPRE is to perform audio preprocessing in the KERAS layers.

Forecasting at scale analyzed by Taylor SJ, Letham B. 2017 where they focused on many iterations of forecasting and they implemented it on Facebook data. They used Facebook data so they applied the PROPHET method which was developed by Facebook. They also used the simple modular regression model which provides the best performance using default parameters. Their main focus was on the flagging forecast for manual check and measure or tracking forecast accuracy. Finally, they concluded that simple and adjustable models help to give better performance with a large amount or a variety of data.

Basically, there are three categories of layer: input, hidden and output. The input layer takes the raw input data as its input; the hidden layers take the output from the previous layer as its input and the output from the output layer are seen as the results of the ANN (Artificial Neural Network). Hidden layers are titled hidden as one does not see or know their inputs or outputs. They are often characterized as feature detectors (Shachmurove, 2002). Generally, it is accepted to have n input perceptron for n inputs however there is no recognized optimum number of hidden perceptron or layers and the number of output perceptron depends on the number of outputs necessary (Kartalopoulos, 1996, Hastie et al., 2001).

"No indication is provided as to the optimal number of nodes per layer. There is no formal method to determine this optimal number; typically one uses trial and error." [Kartalopoulos, 1996]

3 RESEARCH AIMS AND OBJECTIVES

Time series forecasting includes creating and utilizing a predictive model by analyzing the historical data which has a relationship between observations. The decision which is made after applying models, directly impact each and every step of project including the evaluation of forecast models to the fundamental difficulties of the forecast research. Stock price forecasting is the demonstration of attempting to decide the future estimation for any companies stock.

Research Question- Comparison of neural network and deep learning algorithms with traditional statistical learning approaches for stock price forecasting by analyzing NSE listed banks stock price.

Aim- Assist Indian banks with decisions relating to stock price forecasting using time series analysis.

Objective- To analyze and compare different deep learning and statistical learning algorithms trained on 15-20 years of bank data for 5 banks to predict the estimated price of stocks.
The objectives are:

1) Model specification (or model identification);
2) Model fitting (or model estimation);
3) Model evaluation (or model accuracy assessment).

3.1 Forecasting Methods

1) Autoregressive integrated moving average (ARIMA) methods
2) KERAS with LSTM (Long Short Term Memory) model
3) PROPHET Algorithm

3.2 Proposed Approach

The CRISP-DM methodology (Devi, B.U., Sundar, D. and Alli, P., 2013) is a 6 steps (Business understanding, Data understanding, and Data preparation, Modeling, Evaluation and Deployment) process for identifying, selecting, and assessing conditional mean models for discrete, Univariate time series data. For this dissertation proposed steps are:

Step 1: Business Data Understanding

1) Business objectives
2) Business importance
3) Sample data collection
4) Describe and explore the data

Step 2: Data Preparation

1) Transform the data to stabilize the attributes
2) Integrate and format data

Step 3: Data Modeling

1) Examine the data to identify potential models
2) Estimation and testing
3) Predict and forecast

Step 4: Data Evaluation

1) Evaluate results
2) Compare models

4 METHODOLOGY

Moving from machine learning to time-series forecasting is a radical change. It is a challenging, yet enriching experience that helped in understanding in a better way how machine learning can be applied to forecast the stock price. The objective of a predictive model was to estimate the value of an unknown variable. A time series has time *(t)* as an independent variable and a target dependent variable *(y).* The output of the model is the predicted value for *y* at time *t(y`).*

Time series models- ARIMA, PROPHET, and KERAS with LSTM were used for stock price prediction. Sample data collected from Yahoo Finance in the CSV (Comma-separated values) format. CRISP-DM methodology concept was applied to design stock price data transformation and load in the Rstudio for model evaluation. All 5 models were applied for 5 banks (HDFC, ICICI, SBI, KOTAK, and AXIS) stock price data to compare the time series algorithms.

4.1 Sample Data Preparation

The data set consists of the stock price for National Stock Exchange listed 5 Indian corporate banks for the period of 20 years. It includes information like date, stock close price, number of shares and market capital. For this work selected companies from banking sectors for study which are HDFC Bank, AXIS Bank, Kotak Bank, SBI Bank and ICICI Bank. These banks were identified by the help of NIFTY 50 index. The data for these banks were extracted from the available data on Yahoo Finance in the .csv format and was subjected to preprocessing to obtain the stock price using the R programming language. Read the data in R to and then converted it into time series *(ts)* format.

```
> Axis=read.csv ("C:/Users/user/Desktop/M. Sc. Data/M.Sc.
Thesis/data1shares/Axis_Monthly_Avg_Share.csv",header=TRUE)

> Train_Axis=ts(Axis$Monthly_Avg_Share_Price,start=c(1999,01), frequency = 12)
```

The work was based on a sliding window approach for a short term future prediction. To make the data consistency converted the dataset into a monthly average. In order to get more accurate result divided dataset for all 5 banks into trainset and test set in order of trainset- 90%, 80%, 70%, 60% and test set- 10%, 20%, 30%, 40% and then used for loop to calculate the accuracy. Below is the example of all 3 models applied in R.

```
> Axis_fit=Arima (Train_Axis,order = c(Least_AIC$p,Least_AIC$d,Least_AIC$q))

> M=prophet (train)
> future <- make_future_dataframe(M, periods = N-n,freq = 'm')

> model <- keras_model_sequential()
> model%>%
  layer_lstm(units, batch_input_shape = c(batch_size, X_shape2, X_shape3), stateful= TRUE)%>%
  layer_dense(units = 1)
```

So result look like below:

	Train	Test	RMSE Values
ARIMA	146 (60%)	97 (40%)	223.8319
	170 (70%)	73 (30%)	216.0632
	194 (80%)	49 (20%)	84.51927
	219 (90%)	24 (10%)	86.21248

Table 4.1: Least RMSE value

Time Series Features Selection
In machine learning features are created either manually or automatically, creating features is one of the most important tasks in applied machine learning which is not there in time series forecasting. This does not mean that features are completely off limits. Instead, they should be used with care because of the following reasons:

- A pure time series model may have similar or even better performance than one using features.

- If the features are predictable then they have some patterns that can help in building a forecast model for each of them. However, keeping in mind that using predicted values as features will propagate the error to the target variable, which may cause higher errors or produce biased forecasts.

4.2 CRISP-DM Design

CRISP-DM (Wirth, R. and Hipp, J., 2000) (Zazzaro, Gaetano & Romano, Gianpaolo & Mercogliano, Paola 2017) (CRoss Industry Standard Process for Data Mining) method is very first and an open standard process that describes a structured approach to plan a data mining project. The CRISP-DM process mainly focuses on making large data mining projects that are more reliable, faster and manageable but less costly.

As a methodology, it includes descriptions of the typical phases of a project, the tasks involved with each phase, and an explanation of the relationships between these tasks and as a process model, CRISP-DM provides an overview of the data mining life cycle.

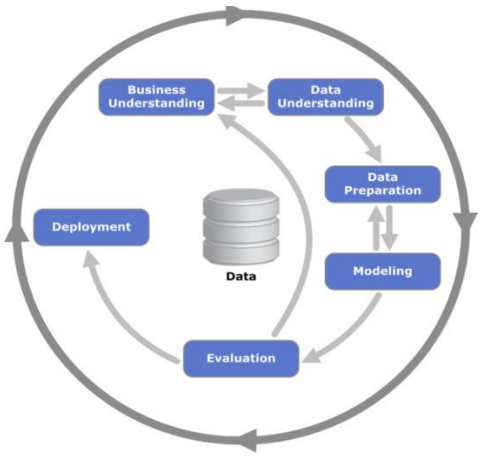


Fig. 4.2: CRISP-DM Methodology

The life cycle model consists of six phases with arrows indicating the most important and frequent dependencies between phases. The CRISP-DM model is flexible and can be customized easily. For example, if an organization aims to detect money laundering, it is likely that it will sift through large amounts of data without a specific modeling goal. Instead of modeling, work will focus on data exploration and visualization to uncover suspicious patterns in financial data. CRISP-DM allows creating a data mining model that fits particular needs.

In such a situation, the modeling, evaluation, and deployment phases might be less relevant than the data understanding and preparation phases. However, it is still important to consider some of the questions raised during these later phases for long-term planning and future data mining goals.

For this thesis deployment phase is not relevant so the methodology used is like CRISM-DM not complete CRISP-DM.

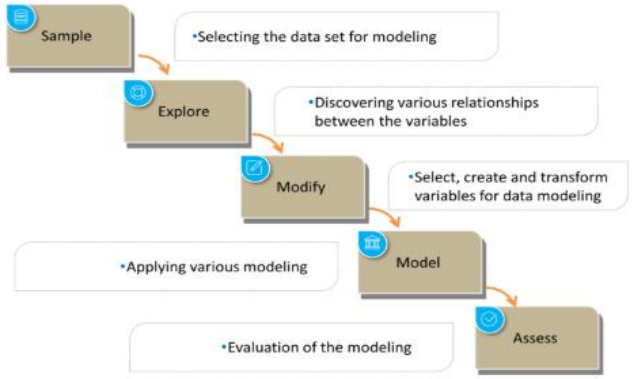


Fig. 4.3: CRISP-DM Methodology like methodology used

4.3 Time Series Components

The factors that are responsible for bringing about changes in a time series, also known as the components of time series, are as follows:

- General Trends
- Seasonal Movements
- Cyclical Movements
- Irregular Fluctuations

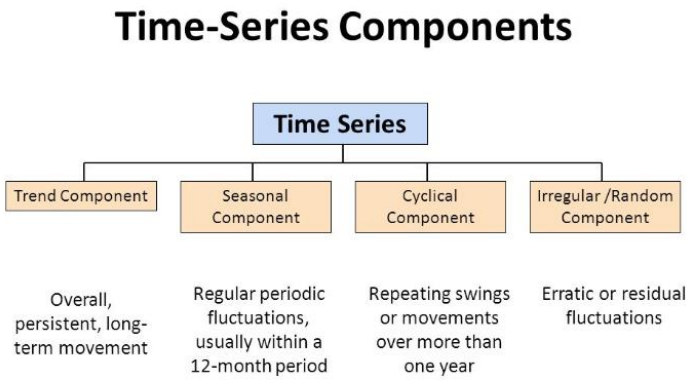


Fig. 4.4 Time Series Components

Trend: A trend exists when a series increases, decreases or remains at a constant level with respect to time.

Seasonality: This property of a time series refers particular periodical patterns that repeat at a constant frequency. Time series models include seasonal variables as dummy features, using binary variables to avoid correlation between features.

Cycles: Cycles are seasons that do not occur at a fixed rate. Also, do not repeat at regular time intervals and may occur even if the frequency is *1*.

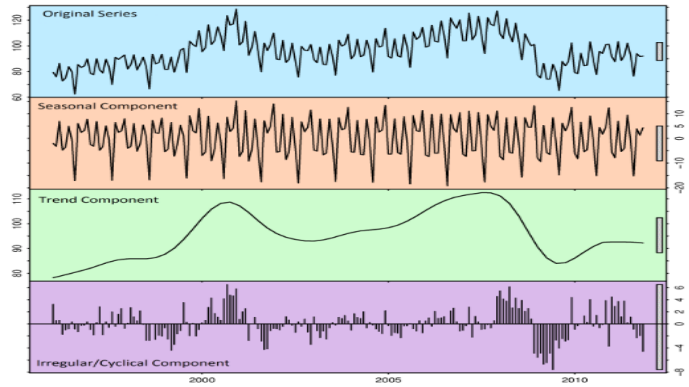


Fig. 4.5 Time Series Components Analysis

Time series components are highly important to analyze variable of interest in order to understand its behavior, what patterns it has, and to be able to choose and fit an appropriate time-series model.

4.4 Procedures and Functions

For this thesis, applied models on training dataset to predict values for the test dataset. The performance evaluation parameter RMSE applied for accuracy measurement after trainset and test set data selection of input and model dependent variables. For seasonal time series, optimized models parameters simultaneously to compare and discuss the result. 3 models are chosen dependent on the basis of the research done on time series forecasting are as below:

4.4.1 ARIMA Model

Since, it is basic to recognize a model (Devi, B.U., Sundar, D. and Alli, P., 2013) to demonstrate the pattern with sufficient data for the financial specialist to settle on a choice. It prescribes that ARIMA is an algorithmic way to deal with change the arrangement is superior to anticipating straightforwardly, and furthermore it gives increasingly exact outcomes. This methodology has excluded any testing criteria for model estimation.

The ARIMA approach was popularized by Box and Jenkins (Devi, B.U., Sundar, D. and Alli, P., 2013), and ARIMA models are often referred to as Box-Jenkins models. The general transfer function model employed by the ARIMA procedure was discussed by Box and Tiao in 1975.

ARIMA model (Auto-Regressive Integrated Moving Average) is a class of statistical models for analyzing and forecasting time series data. It explicitly caters to a suite of standard structures in time series data. The AR part of ARIMA indicates that the evolving variable of interest is regressed on its prior values. The MA part indicates that the regression error is actually a linear combination of error terms whose values occurred contemporaneously and at various times in the past.

Non-seasonal ARIMA models are generally denoted ARIMA (p, d, q) where parameters p, d, and q are non-negative integers, p is the order (number of time lags) of the autoregressive model, d is the degree of differencing (the number of times the data have had past values subtracted), and q is the order of the moving-average model. Seasonal ARIMA models are usually denoted ARIMA(p, d, q) (P, D, Q)m, where m refers to the number of periods in each season, and the uppercase P, D, Q refers to the autoregressive, differencing, and moving average terms for the seasonal part of the ARIMA model.

ARIMA algorithm in three steps:

Step 1: Model identification
Step 2: Model estimation
Step 3: Forecasting

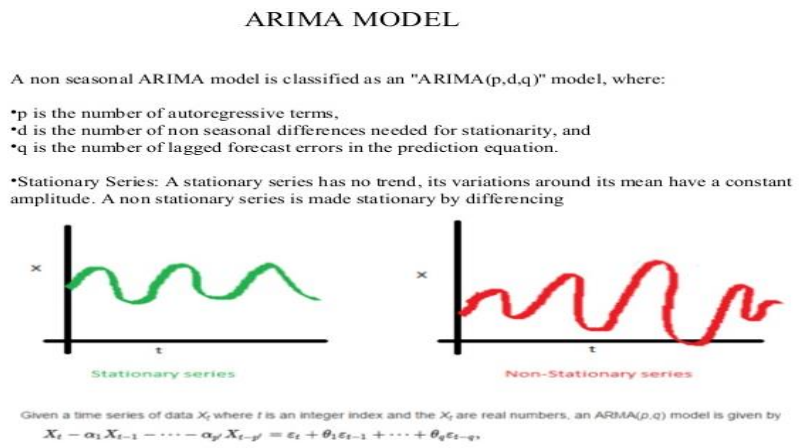


Fig. 4.6 ARIMA model (Time Series)

In an ARIMA model (Pai, P.F. and Lin, C.S., 2005), the future value of a variable is supposed to be a linear combination of past values and past errors. In general, ARIMA model is denoted by ARMA (p, q). The form of the ARMA (p, q) model is,

$$y_t = C + \emptyset_1 y_{t-1} + \emptyset_2 y_{t-2} + \ldots + \emptyset_p y_{t-p} + \in_t + \theta_1 \in_{t-1} + \emptyset_2 \in_{t-2} + \ldots + \theta_q \in_{t-q}$$

Where $\in_t$ is an uncorrelated innovation process with mean zero and y_t is the actual value and $\in_t$ is the random error at time *t*, $\emptyset_1$ and $\emptyset_2$ are the coefficients, *p* and *q* are integers that are often referred to as autoregressive and moving average polynomials.

For example, the ARIMA *(1,0,1)* model can be represented as follows

$$y_t = \theta_0 + \emptyset_1 y_{t-1} + \varepsilon_t - \theta_1 \varepsilon_{t-1}$$

Here, applied for loop using (p,d,q) values between 0 to 2 to find least AIC value which later helped to find best the RMSE value as an outcome.

```
> modelAIC <- data.frame()
> for (d in 0:1)
+ {
+   for (p in 0:2)
+  {
+    for (q in 0:2)
+   {
+       Axis_fit=Arima(Train_Axis,order = c(p,d,q),method="ML")
+       AIC=AIC(Axis_fit)
+       modelAIC <- rbind(modelAIC, data.frame(p,d,q,AIC))
+   }
+  }
+ }
```

After applying for loop, found 54 values in total and 18 combinations for (p, d, q) and 18 AIC values for each bank shown in below figure. Out of 18 AIC values the least AIC values' (p, d, q)

value is (0, 1, 1) (highlighted in yellow). Fig 4.8 represented the graphical visualization for least (p, d, q) which is (0, 1, 1) (acf and partial acf).

p	d	q	AIC
0	0	0	2378.636
0	0	1	2144.671
0	0	2	1984.136
1	0	0	1650.315
1	0	1	1648.608
1	0	2	1649.329
2	0	0	1649.188
2	0	1	1648.483
2	0	2	1650.446
0	1	0	1633.775
0	**1**	**1**	**1632.266**
0	1	2	1632.844
1	1	0	1632.839
1	1	1	1632.033
1	1	2	1633.977
2	1	0	1633.753
2	1	1	1633.963
2	1	2	1635.785

Table 4.7 Least AIC value for ICICI bank

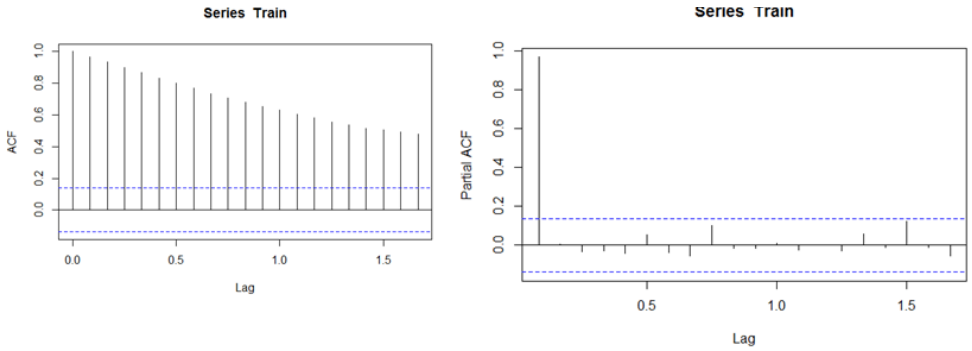


Fig 4.8 Graphical representation of least AIC values (acf, pacf)

4.4.2 PROPHET Forecasting Model

Prophet (Taylor SJ, Letham B. 2017) is a methodology for forecasting time series data dependent on an added substance model where non-direct patterns are fit with yearly, week by week, and daily trends, in addition to occasion impacts. It works best with time series that have solid occasional impacts and a few periods of authentic data. Prophet is vigorous to missing data and moves in the pattern, and commonly handles anomalies well.

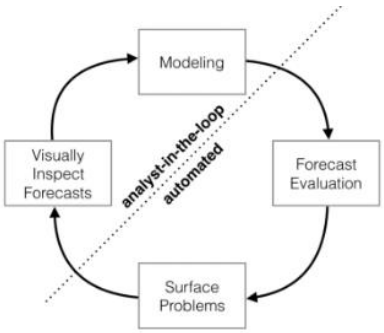


Fig. 4.9 Prophet forecasting flow

Prophet is an additive model with the following components:

$$y(t) = g(t) + s(t) + h(t) + \epsilon_t$$

Where, $g(t)$ model trend, which describes long-term increase or decrease in the data. Prophet incorporates two trend models, a saturating growth model, and a piecewise linear model, depending on the type of forecasting problem.

$s(t)$ Model seasonality with Fourier series, which describes how data is affected by seasonal factors such as the time of the year

$h(t)$ Model the effects of holidays or large events that impact business time series

ϵ_t Represents irreducible error term of any idiosyncratic changes which are not accommodated by the model

Prophet (Taylor SJ, Letham B. 2017) naturally assesses estimate execution and banners issues that warrant manual intercession. One of the most straightforward evaluation methods is to set a benchmark with some basic forecasting techniques. It is helpful to contrast shortsighted and propelled forecasting techniques with decide if extra execution can be picked up by utilizing an increasingly unpredictable model.

The Prophet (Taylor SJ, Letham B. 2017) forecast can predict both the week after week and yearly seasonality's, and dissimilar to the baselines, does not overcompensate to the occasion plunge in the principal year. In the principal forecast, the Prophet model has somewhat over fit the yearly regularity allowed just a single year of information.

In the Prophet model (Taylor SJ, Letham B. 2017) one of the important specification which says there are a few spots where analysts can modify the model to apply their ability and outer information and it is not required to have any comprehension of the fundamental statistics. Some of the key specification to highlight below:

Capacities: Experts can use any outer information from any source for the absolute market estimate and then resourceful information as learning can be applied legitimately by determining limits.

Changepoints: Known dates of change points, such as dates of product changes, can easily be specified.

Holidays and seasonality: Experts' experience involvement help with which occasions sway development in which districts and they can legitimately use as input to store relevant occasion dates and the relevant time sizes of regularity or seasonality.

Smoothing parameters: Hereby changing the value of τ, this can be chosen from inside scope of increasingly worldwide or locally smooth models. The seasonality and holiday smoothing parameters *(σ, ν)* help model to amount of the historical seasonal variety which will be expected in the future.

In order to get the best predicted values used PROPHET package in R with trainset (60%, 70%, 80%, and 90%) and test set (40%, 30%, 20% and 10%). Below is the R code:

```
>i=0
> M=prophet(train)
> future <- make_future_dataframe(M, periods = N-n,freq = 'm')
> FC= predict(M, future)
> Pred_value=list(0)
```

```
> Actual_value=list(0)

> for (x in c(0.6,0.7,0.8,0.9))
+ {
+ N = nrow(df)
+ n = round(N *x)
+ train = df[1:n, ]
+ test  = df[(n+1):N, ]

+ M= prophet(train)
+ future<- make_future_dataframe(M, periods = N-n,freq = 'm')
+ FC= predict(M, future)
+ }
```

Below table shows the result of the above query which represented predicted values for test dataset (40%, 30%, 20%, and 10%) for PROPHET model. Similarly, result tables have been generated for all 5 banks after applying PROPHET model.

Pred values_40	Pred values_30	Pred values_20	Pred values_10
237.6298468	275.7105767	341.2668104	544.2299788
245.1902297	279.5206584	351.1294229	553.8058583
253.7697587	279.7811739	357.7469229	557.1304658
255.5605151	279.1948909	361.0561604	564.0741909
260.4592625	283.7491245	365.3562378	565.788838
261.1660298	285.2513247	360.3097042	569.4520383
265.5374135	288.2537366	363.1235096	572.6965856
270.9613008	293.8515743	368.8047127	573.9355997
273.5315588	294.7924521	376.3092254	577.4671241
274.4772095	295.6082705	377.4894395	584.576187

Table 4.10 Predicted values after applying Prophet Model

4.4.3 KERAS with LSTM model

The LSTM (Long short-term memory) (Gulli, A. and Pal, S., 2017) is a variant of RNN that is capable of learning long term dependencies. LSTMs were first proposed by Hochreiter and Schmidhuber and refined by many other researchers. They work well on a large variety of problems and are the most widely used type of RNN. In deep learning, while preprocessing data using KERAS (Choi, K., Joo, D. and Kim, J., 2017) frequently involves bunches of time and effort.

The Simple RNN (Gulli, A. and Pal, S., 2017) uses the hidden state from the previous time step and the current input in a *tanh* layer to implement recurrence. LSTMs also implement recurrence in a similar way, but instead of a single tanh layer, there are four layers interacting in a very specific way.

The following diagram (4.11) illustrated the transformations that are applied to the hidden state at time step *t* where *c* is cell state (top line) represents the internal memory of the unit and *i*, *f*, *o*, and *g* are gates for a hidden state (bottom line):

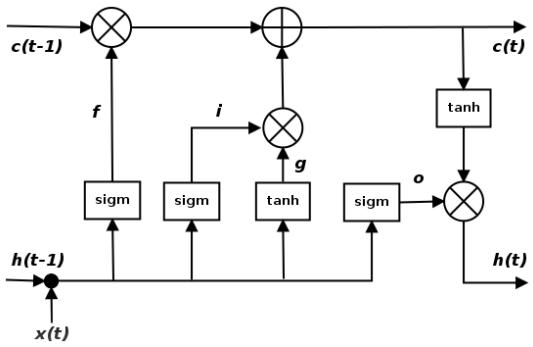


Fig. 4.11 RNN transformation using hidden state

The LSTM (Long short-term memory) (Gulli, A. and Pal, S., 2017) is an artificial recurrent neural network (RNN) architecture used in the field of deep learning. The core data structure of KERAS is a model, a way to organize layers. The simplest type of model is the Sequential model, a linear stack of layers. For more complex architectures the KERAS functional API can be used which allow building arbitrary graphs of layers.

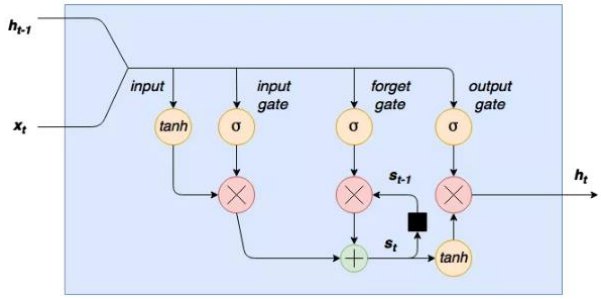


Fig. 4.12 LSTM cell diagram

The mathematics of the LSTM cell looks like this:
So LSTM cell takes the previous memory state C_{t-1} and does element-wise multiplication with forget gate *(f)*

$$C_t = C_t * f_t$$

If forget gate value is 0 then previous memory state is completely forgotten

If forget gate value is 1 then previous memory state is completely passed to the cell (where *f* gate gives values between *0* and *1*)

Now with current memory stateC_t, we calculate new memory state from input state and C layer.

$$C_t = C_t + (I_t * C`_t)$$

C_t = Current memory state at time step t. and it gets passed to next time step.

Finally, the output will be based on cell state C_t but will be a filtered version. So need to apply *Tanh* to C_t then do element-wise multiplication with the output gate *O*, Which will be current hidden state H_t

$$H_t = Tanh\,(C_t)$$

Then pass these two C_t and H_t to the next time step and repeat the same process.

In order to get best future predicted values used KERAS package with LSTM for trainset (60%, 70%, 80%, and 90%) and test set (40%, 30%, 20% and 10%).

```
> model <- keras_model_sequential()
> model%>%
+  layer_lstm(units, batch_input_shape = c(batch_size, X_shape2, X_shape3), s+ tateful= TRUE)%>%
+ layer_dense(units = 1)

# Compiile
> model %>% compile(
+ loss = 'mean_squared_error',
+ optimizer = optimizer_adam( lr= 0.02, decay = 1e-6 ),
+ metrics = c('accuracy') )

# Fit the model
> Epochs = 50
> for(i in 1:Epochs )
+ {
+ model %>% fit(x_train, y_train, epochs=1, batch_size=batch_size, verbose=1,+ shuffle=FALSE)
+ model %>% reset_states()
+ }
```

Below table shows the result of the above query which represented predicted values for test dataset (40%, 30%, 20%, and 10%) for KERAS with LSTM model. Similarly, result tables have been generated for all 5 banks after applying KERAS with LSTM model.

Pred values_40	Pred values_30	Pred values_20	Pred values_10
247.3459959	291.1640218	596.762059	509.6231128
267.7419943	264.1245641	552.026026	507.1549421
287.941526	269.8529823	576.0533667	512.7737913
244.0223741	305.0951613	567.533313	512.1576064
253.447822	252.0501557	597.4809896	527.3018267
265.1832621	240.9404837	547.7621889	499.7612001
232.1572795	189.7457521	494.7430944	519.9194316
218.5155607	203.8292693	514.9160381	499.1905598

Table 4.13 Future predicted values after applying KERAS package with LSTM

4.5 Software and Packages

R is a programming language and free software environment for statistical computing and graphics supported by the R Foundation for Statistical Computing.

R is an integrated suite of software facilities for data manipulation, calculation, and graphical display. It is a well-developed, simple and effective programming language which includes conditionals, loops; user defined recursive functions and input, and output facilities.
R programming is a strong foundation in functional programming. The ideas of functional programming are well suited which help in solving many of the challenges for data analysis. R provides a powerful and flexible toolkit which allows writing concise yet descriptive code.

Contrasted with other programming languages, the R language, in general, is increasingly focused on results rather than procedures. Information of programming designing accepted procedures is inconsistent. R is very useful assets for imparting outcomes. R packages make it simple to use or deliver HTML or pdf reports or create interactive websites.

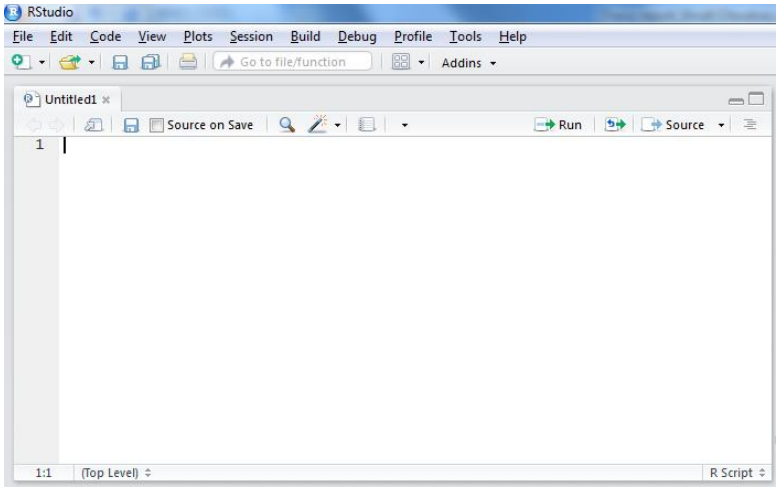


Fig 4.14 Rstudio console

Packages and Libraries: install.packages('prophet')
install.packages('rsample')
install.packages('tensorflow')
library(sqldf)
library(tidyverse)
library(keras)
library(forecast)
library(prophet)
library(data.table)
library(dplyr)
library(ggplot2)

Number of Files: 5 csv files (AXISBANK.csv, ICICIBANK.csv, HDFCBANK.csv, KOTAKBANK.csv, SBIBANK.csv)

Number of records: Each csv file contains about 20 years' stock price data which has approx 5000 records each.

5 DATA ANALYSIS

5.1 Exploratory Data Analysis

Exploratory data analysis (EDA) is a systematic way to explore the data using transformation and visualization. EDA is an iterative cycle and it's not a process with any set of rules however some basic steps to be applied that help to manage data in a systematic way:

Step 1: Generate questions about data.

Step 2: Search for answers by visualizing, transforming, and modeling the data.

Step 3: Use what is to learn to refine questions and if required then generate new questions.

5.1.1 Sample Data Analysis

After sample data preparation the next task in modeling was sample data analysis and visualization. Fig (5.1) and (5.2) representing the monthly average share price for ICICI and SBI bank. Both graphs were a representation of stock closing price index for almost 20-25 years from 2002 to 2019 for ICICI bank and 1996 to 2019 for SBI bank. X-axis represented years-months and y-axis represented an average of each month stock closing price.

There were up and down trends observed many times during the last 20 years. The stock price has experienced continuous growth since 2002. There was a sharp fall observed between 2008-2009, similarly, in Dec-2016, a sharp fall was observed. There are multiple sharp peaks observed for multiple years for ICICI bank, which could be a sign of good growth or a stock bubble.

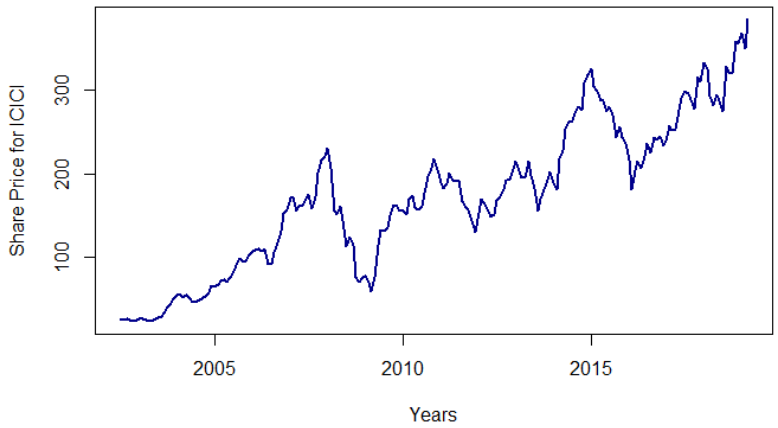


Fig 5.1 Graphical representation of ICICI bank stock closing price index

The SBI bank graph was a representation of stock closing price index for almost 25 years from 1996 to 2019. X-axis represented years-months and y-axis represented an average of each month stock closing price. There were up and down trends observed many times during the last 25 years. The stock price has experienced very slow growth since 1996. After 2007 SBI bank stock price started growing usual than expected then there was a sharp fall during 2010. There are multiple sharp peaks observed for multiple years, which could be a sign of good growth or a stock bubble.

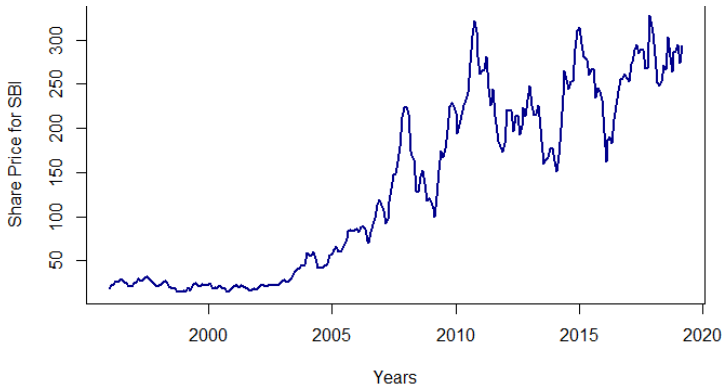


Fig 5.2 Graphical representation of SBI bank stock closing price index

Until now data was non-stationary and it is required to have stationary data for time series forecasting because it makes future prediction easier with stationary series data. For stationary time series, it is required to have the mean and variance constant over time. The perfect way to check whether mean and variance are constant or not is plotting which helps to find the correct difference for time series. Below fig shows non-stationary data.

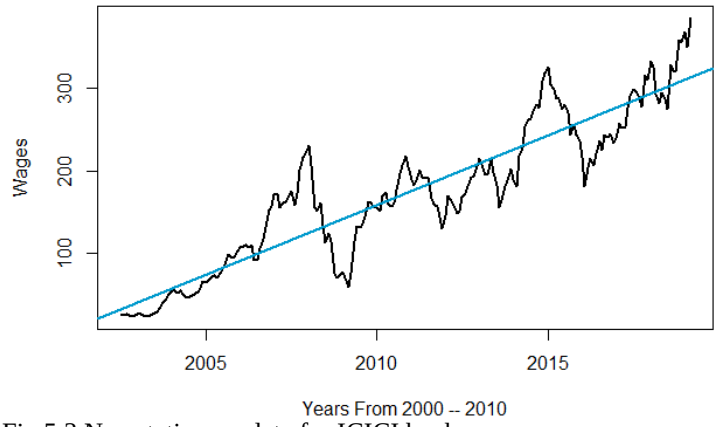


Fig 5.3 Non stationary data for ICICI bank

To make data stationary differencing method is used which helps data to transform from non-stationary to stationary time series. Validation of the assumptions has been done using graphical visualization. The difference between the current time and previous time period is first differencing value. After applying differencing method time series looked like below which is stationary data.

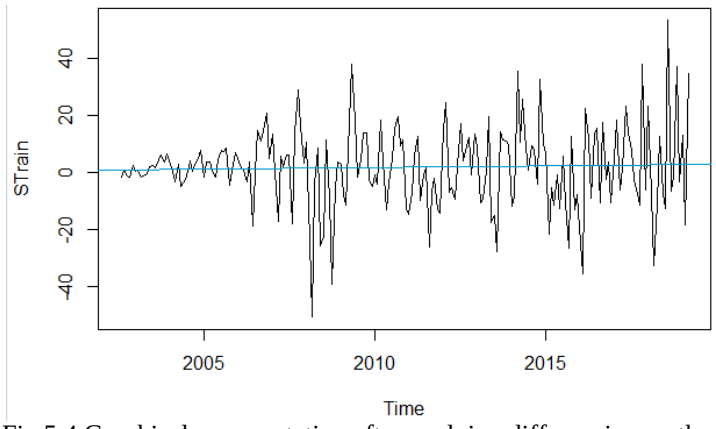


Fig 5.4 Graphical representation after applying differencing method

5.1.2 RMSE value analysis

Root Mean Square Error (RMSE) calculates the error between population values predicted by a model and the values observed. RMSE is a proportion of accuracy which helps to analyze forecasting errors of various models for a specific dataset since it is scale-subordinate.

Once the sample data analysis was completed then the next step was an evaluation of the models. Here, 3 models were applied which were ARIMA, PROPHET, and KERAS with LSTM. Models result has been analyzed by RMSE value. Fig (5.5) (5.6) (5.7) (5.8) show the RMSE values for all 3 models for all 5 banks.

Since there were 4 train dataset (60%, 70%, 80%, 90%) and test dataset (40%, 30%, 20%, 10%) it got easier to find out best performance for all 3 models. The table represented for Axis bank (Fig 5.5). The ARIMA model was showing the lowest RMSE value for 20% test data evaluation and higher for 40% test data prediction. Similarly, PROPHET model has shown the lowest RMSE value for 10% test data evaluation and higher for 30% test data prediction. And KERAS with LSTM model has shown the lowest RMSE value for 40% test data evaluation and higher for 20% test data prediction. (Lowest RMSE value highlighted with red color)

	Train	Test	RMSE Values
ARIMA	146 (60%)	97 (40%)	223.8319
	170 (70%)	73 (30%)	216.0632
	194 (80%)	49 (20%)	84.51927
	219 (90%)	24 (10%)	86.21248
	Train	Test	RMSE Values
PROPHET	146 (60%)	97 (40%)	85.4952291
	170 (70%)	73 (30%)	139.4265467
	194 (80%)	49 (20%)	132.0441463
	219 (90%)	24 (10%)	57.17100414
	Train	Test	RMSE Values
LSTM	146 (60%)	97 (40%)	6.309496112
	170 (70%)	73 (30%)	7.627314237
	194 (80%)	49 (20%)	12.53491778
	219 (90%)	24 (10%)	8.177643154

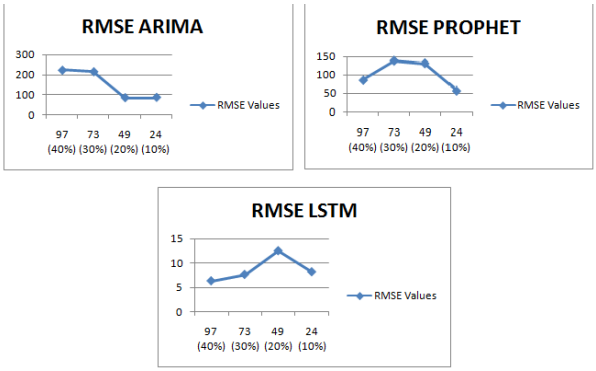


Fig 5.5 RMSE values analysis for AXIS bank for all 3 models

The table represented for HDFC bank (Fig 5.6). The ARIMA model was showing the lowest RMSE value for 10% test data evaluation and higher for 40% test data prediction. Similarly, PROPHET model has shown the lowest RMSE value for 10% test data evaluation and higher for 40% test data prediction. And KERAS with LSTM model has shown the lowest RMSE value for 20% test data evaluation and higher for 30% test data prediction. (Lowest RMSE value highlighted with red color)

	Train	Test	RMSE Values
ARIMA	167 (60%)	112 (40%)	858.1191
	195 (70%)	84 (30%)	852.5168
	223 (80%)	56 (20%)	520.7688
	251 (90%)	28 (10%)	511.3367
	Train	Test	RMSE Values
PROPHET	167 (60%)	112 (40%)	687.9130796
	195 (70%)	84 (30%)	589.8793149
	223 (80%)	56 (20%)	574.7622151
	251 (90%)	28 (10%)	465.5449221
	Train	Test	RMSE Values
LSTM	167 (60%)	112 (40%)	15.92855488
	195 (70%)	84 (30%)	18.06708961
	223 (80%)	56 (20%)	10.27311767
	251 (90%)	28 (10%)	15.84698092

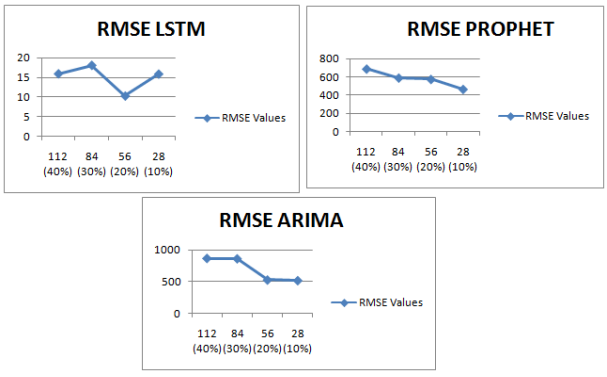
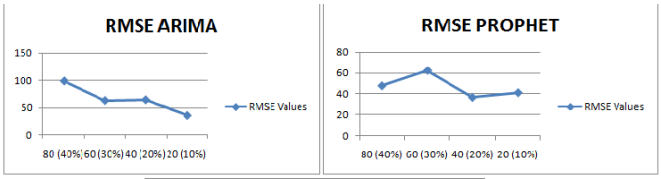


Fig 5.6 RMSE values analysis for HDFC bank for all 3 models

The table represented for ICICI bank (Fig 5.7). The ARIMA model was showing the lowest RMSE value for 10% test data evaluation and higher for 40% test data prediction. Similarly, PROPHET model has shown the lowest RMSE value for 20% test data evaluation and higher for 30% test data prediction. And KERAS with LSTM model has shown the lowest RMSE value for 10% test data evaluation and higher for 40% test data prediction. (Lowest RMSE value highlighted with red color)

	Train	Test	RMSE Values
ARIMA	121 (60%)	80 (40%)	99.1891
	141 (70%)	60 (30%)	63.35003
	161 (80%)	40 (20%)	64.74696
	181 (90%)	20 (10%)	36.43887
	Train	Test	RMSE Values
PROPHET	121 (60%)	80 (40%)	47.59108218
	141 (70%)	60 (30%)	62.01689325
	161 (80%)	40 (20%)	36.72500746
	181 (90%)	20 (10%)	41.16325737
	Train	Test	RMSE Values
LSTM	121 (60%)	80 (40%)	7.020169962
	141 (70%)	60 (30%)	6.423424443
	161 (80%)	40 (20%)	6.561318474
	181 (90%)	20 (10%)	3.184659373



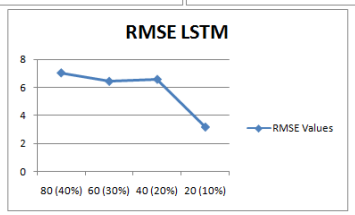
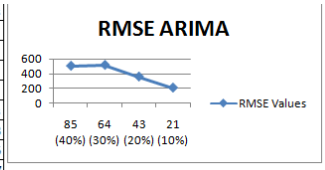
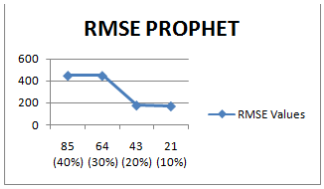


Fig 5.7 RMSE values analysis for ICICI bank for all 3 models

The table represented for KOTAK bank (Fig 5.8). The ARIMA model was showing the lowest RMSE value for 10% test data evaluation and higher for 30% test data prediction. Similarly, PROPHET model has shown the lowest RMSE value for 10% test data evaluation and higher for 40% test data prediction. And KERAS with LSTM model has shown the lowest RMSE value for 40% test data evaluation and higher for 10% test data prediction. (Lowest RMSE value highlighted with red color)

	Train	Test	RMSE Values
ARIMA	128 (60%)	85 (40%)	507.9019
	149 (70%)	64 (30%)	519.1905
	170 (80%)	43 (20%)	359.1787
	192 (90%)	21 (10%)	207.6787
	Train	Test	RMSE Values
PROPHET	128 (60%)	85 (40%)	449.4291813
	149 (70%)	64 (30%)	448.0328095
	170 (80%)	43 (20%)	176.9677967
	192 (90%)	21 (10%)	169.3145037
	Train	Test	RMSE Values
LSTM	128 (60%)	85 (40%)	6.420621576
	149 (70%)	64 (30%)	9.610269982
	170 (80%)	43 (20%)	24.2082226
	192 (90%)	21 (10%)	15.03665357





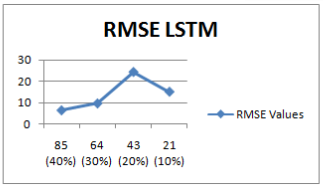
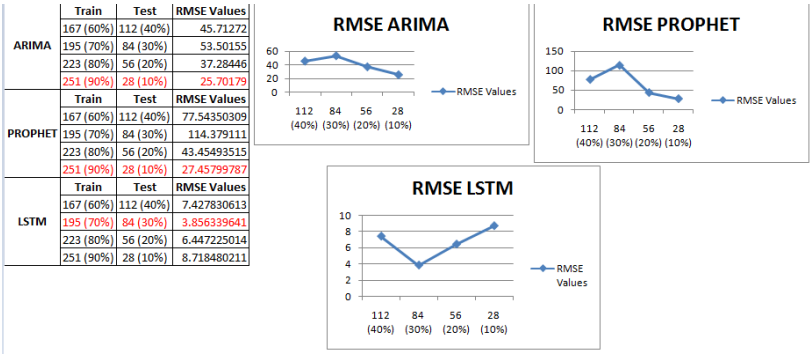


Fig 5.8 RMSE values analysis for KOTAK bank for all 3 models

The table represented for SBI bank (Fig 5.9). The ARIMA model was showing the lowest RMSE value for 10% test data evaluation and higher for 40% test data prediction. Similarly, PROPHET model has shown the lowest RMSE value for 10% test data evaluation and higher for 40% test data prediction. And KERAS with LSTM model has shown the lowest RMSE value for 30% test data evaluation and higher for 10% test data prediction. (Lowest RMSE value highlighted with red color)



	Train	Test	RMSE Values
ARIMA	167 (60%)	112 (40%)	45.71272
	195 (70%)	84 (30%)	53.50155
	223 (80%)	56 (20%)	37.28446
	251 (90%)	28 (10%)	25.70179
	Train	Test	RMSE Values
PROPHET	167 (60%)	112 (40%)	77.54350309
	195 (70%)	84 (30%)	114.379111
	223 (80%)	56 (20%)	43.45493515
	251 (90%)	28 (10%)	27.45799787
	Train	Test	RMSE Values
LSTM	167 (60%)	112 (40%)	7.427830613
	195 (70%)	84 (30%)	3.856339641
	223 (80%)	56 (20%)	6.447225014
	251 (90%)	28 (10%)	8.718480211

Fig 5.9 RMSE values analysis for SBI bank for all 3 models

5.1.3 20% Test Forecasting Analysis

After analyzing RMSE values it was found that most of the models have given the best performance for 20% and 10% test dataset. Hence for the next step which is forecasting, therefore, performed forecasting analysis on 20% and 10% test dataset. Out of 5 banks discussed 2 banks SBI and ICICI bank.

Fig (5.10) (5.11) was shown forecasting for 20% test dataset for SBI and ICICI bank. All 3 models have been drawn in the graph. Red line represents actual forecasting, the blue dotted line represents the PROPHET model, the green line represents the LSTM model and the black line represents the ARIMA model. PROPHET model had shown straight line for forecasting for 20% test dataset whereas LSTM and ARIMA models had shown some trends and patterns for both the banks.

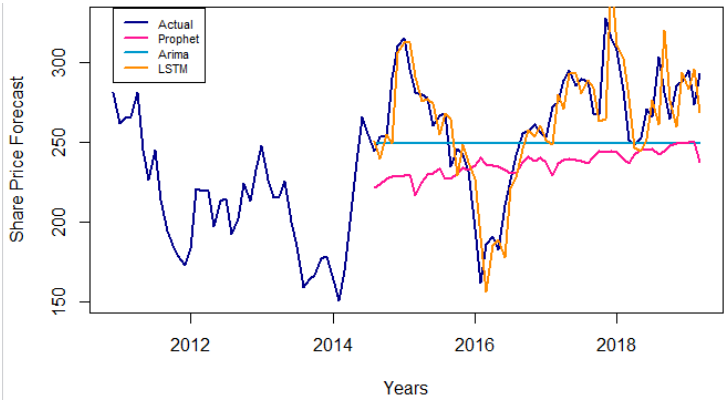


Fig 5.10 SBI bank stock price forecasting for 20% test dataset

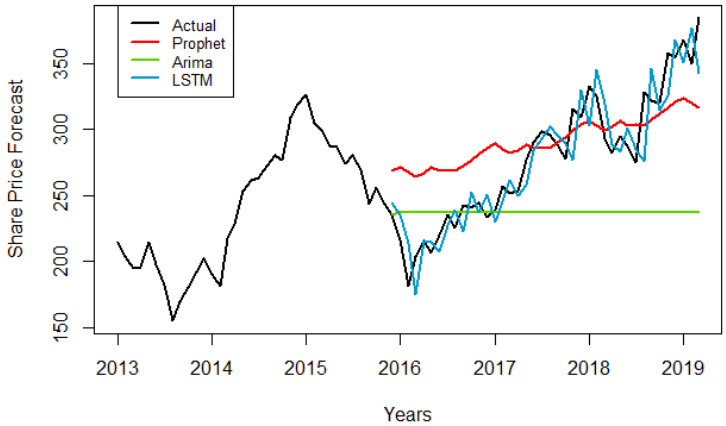


Fig 5.11 ICICI bank stock price forecasting for 20% test dataset

5.1.4 10% Forecasting Analysis

Fig (5.12) (5.13) was shown forecasting for 10% test dataset for SBI and ICICI bank. All 3 models have been drawn in the graph. Red line represents actual forecasting, the blue dotted line represents the PROPHET model, the green line represents the LSTM model and the black line represents the ARIMA model. ARIMA model had shown straight line for forecasting for 10% test dataset whereas LSTM and PROPHET models had shown some trends and patterns for both the banks.

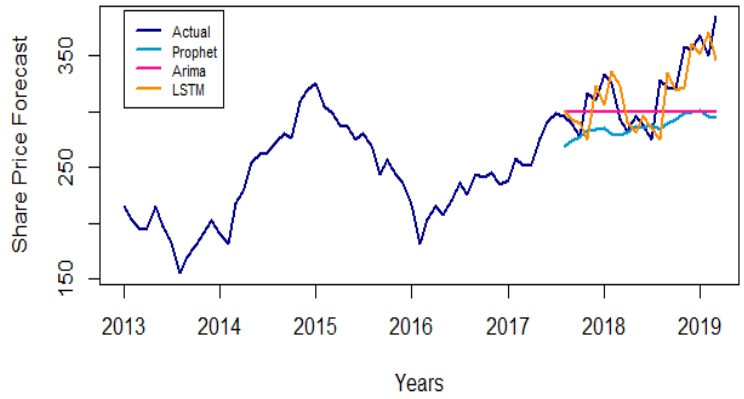


Fig 5.12 SBI bank stock price forecasting for 10% test dataset

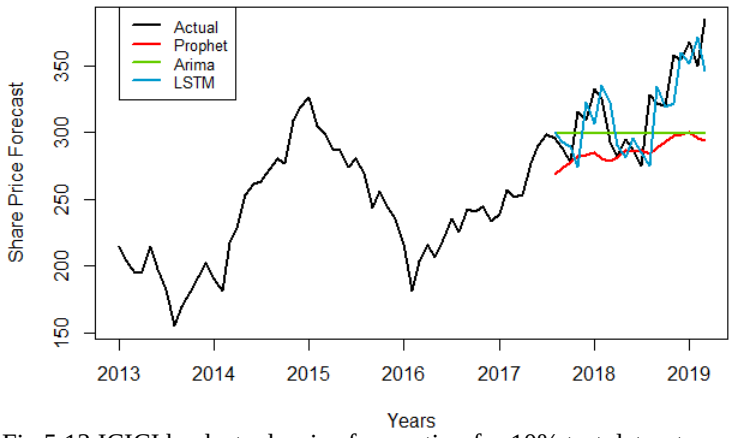


Fig 5.13 ICICI bank stock price forecasting for 10% test dataset

The summarized outcome of this thesis is as below:

1) The performance of each algorithm varies with a different sample dataset.

2) The monthly average accuracy needs to be calculated for each model to predict accurately for all the banks.

3) RMSE is used for measurement as it is a measure which is easy to interpret as well as precise.

4) Backtesting proved to be an effective technique to test the accuracy of Time series prediction models for all datasets.

6 RESULTS AND DISCUSSION

6.1 Result Review

The ARIMA model has been picked for a single time series ought not to change that much and pursue some pattern for order selection. The pattern segment of PROPHET was important to the point that it delivered very nearly a straight line that's the reason why it didn't diminish auspiciously. The LSTM prediction depended on a lot of last qualities, along these lines less inclined to change because of seasonality and the current pattern.

There were 3 methods (ARIMA, PHOPHET, KERAS with LSTM) applied for 5 Indian banks (AXIS, HDFC, ICICI, KOTAK, and SBI). The performance evaluation parameter RMSE applied after trainset and test set data selection of input and model dependent variables. For seasonal time series, optimized all 3 model parameters simultaneously to compare and discuss the result. Let's review results for all 3 models.

6.1.1 ARIMA (Autoregressive Moving Integrated Average model)

The ARIMA model had been implemented for 5 Indian banks using train dataset and least AIC (Akaike Information Critera) value has been calculated to find the accuracy. Least AIC value is highlighted in yellow in the below figure. Train dataset has been divided in such a way (60%, 70%, 80%, and 90%) so that the best result can be evaluated. Accuracy (RMSE) was calculated by the error between population values predicted by a model and the values observed.

p	d	q	AIC
0	0	0	3267.372
0	0	1	2964.869
0	0	2	2723.835
1	0	0	2137.802
1	0	1	2120.599
1	0	2	2122.283
2	0	0	2119.469
2	0	1	2121.485
2	0	2	2122.373
0	1	0	2119.393
0	1	1	2102.322
0	1	2	2103.746
1	**1**	**0**	**2101.26**
1	1	1	2103.255
1	1	2	2105.051
2	1	0	2103.257
2	1	1	2105.256
2	1	2	2106.687

p	d	q	AIC
0	0	0	4333.727
0	0	1	3966.836
0	0	2	3679.36
1	0	0	2672.85
1	0	1	2622.58
1	0	2	2617.071
2	0	0	2946.133
2	0	1	2624.517
2	0	2	2617.689
0	1	0	2652.062
0	1	1	2601.781
0	1	2	2603.703
1	1	0	2607.591
1	1	1	2603.613
1	1	2	2586.173
2	1	0	2607.347
2	1	1	2602.16
2	**1**	**2**	**2583.96**

p	d	q	AIC
0	0	0	2378.636
0	0	1	2144.671
0	0	2	1984.136
1	0	0	1650.315
1	0	1	1648.608
1	0	2	1649.329
2	0	0	1649.188
2	0	1	1648.483
2	0	2	1650.446
0	1	0	1633.775
0	**1**	**1**	**1632.266**
0	1	2	1632.844
1	1	0	1632.839
1	1	1	1632.033
1	1	2	1633.977
2	1	0	1633.753
2	1	1	1633.963
2	1	2	1635.785

p	d	q	AIC
0	0	0	3114.492
0	0	1	2839.911
0	0	2	2598.929
1	0	0	1995.413
1	0	1	1951.562
1	0	2	1952.444
2	0	0	1956.09
2	0	1	1952.492
2	0	2	1948.622
0	1	0	1975.757
0	**1**	**1**	**1931.997**
0	1	2	1932.817
1	1	0	1936.604
1	1	1	1932.958
1	1	2	1934.764
2	1	0	1932.689
2	1	1	1934.627
2	1	2	1936.543

p	d	q	AIC
0	0	0	3377.262
0	0	1	3014.891
0	0	2	2791.64
1	0	0	2253.5
1	0	1	2236.807
1	0	2	2237.94
2	0	0	2239.628
2	0	1	2238.778
2	0	2	2231.787
0	1	0	2237.884
0	1	1	2221.71
0	1	2	2222.634
1	1	0	2224.645
1	1	1	2219.313
1	**1**	**2**	**2216.611**
2	1	0	2223.395
2	1	1	2218.095
2	1	2	2218.57

Table 6.1 Least AIC value for all 5 banks (AXIS, HDFC, ICICI, KOTAK and SBI)

Here, applied for loop in R programming language using (p, d, q) values between 0 to 2 to find the least AIC value which later helped to find the best RMSE value as an outcome. As per for result, least AIC value has been measured which are for AXIS bank least AIC values are (1, 1, 0), for HDFC bank least AIC values are (2, 1, 2), for ICICI bank is least AIC values are (0, 1, 1), KOTAK bank is (0, 1, 1) and SBI bank is (1, 1, 2). To evaluate accuracy (RMSE) value applied the ARIMA model using these (p, d, q) values.

Accuracy has also been calculated using 4 sets of train dataset and test dataset. Below fig has shown the RMSE value for all 5 banks for 4 sets. Least RMSE value for AXIS bank is 84.52 (80% trainset and 20% test set), HDFC bank is 511.34 (90% trainset and 10% test set), ICICI bank is 36.44 (90% trainset and 10% test set), KOTAK bank is 207.68 (90% trainset and 10% test set) and SBI bank is 25.70 (90% trainset and 10% test set).

Train	Test	RMSE Values
146 (60%)	97 (40%)	223.8319
170 (70%)	73 (30%)	216.0632
194 (80%)	49 (20%)	84.51927
219 (90%)	24 (10%)	86.21248

Train	Test	RMSE Values
167 (60%)	112 (40%)	858.1191
195 (70%)	84 (30%)	852.5168
223 (80%)	56 (20%)	520.7688
251 (90%)	28 (10%)	511.3367

Train	Test	RMSE Values
121 (60%)	80 (40%)	99.1891
141 (70%)	60 (30%)	63.35003
161 (80%)	40 (20%)	64.74696
181 (90%)	20 (10%)	36.43887

Train	Test	RMSE Values
128 (60%)	85 (40%)	507.9019
149 (70%)	64 (30%)	519.1905
170 (80%)	43 (20%)	359.1787
192 (90%)	21 (10%)	207.6787

Train	Test	RMSE Values
167 (60%)	112 (40%)	45.71272
195 (70%)	84 (30%)	53.50155
223 (80%)	56 (20%)	37.28446
251 (90%)	28 (10%)	25.70179

Table 6.2 Accuracy (RMSE) values for all 5 banks (AXIS, HDFC, ICICI, KOTAK and SBI)

Below fig has shown the graphical representation of RMSE values for all 5 banks where least RMSE value can easily be identified after applying the ARIMA model.

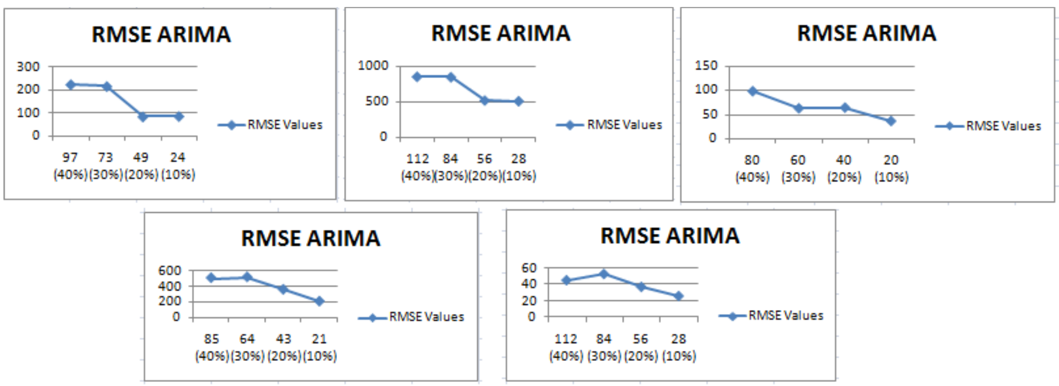


Fig 6.3 Graphical representation of RMSE values for all 5 banks (AXIS, HDFC, ICICI, KOTAK and SBI)

Below fig has shown the predicted values for all 5 banks after applying the ARIMA model for 4 trainset and test set.

Pred values_40	Pred values_30	Pred values_20	Pred values_10
244.7711139	290.3707334	578.1434178	507.9400058
243.3057293	291.1114148	581.9852151	509.2488949
242.7102989	291.3378099	583.2803481	509.7359743
242.4683574	291.4070094	583.7169587	509.917232
242.3700491	291.4281607	583.8641473	509.9846838
242.3301035	291.4346258	583.913767	510.0097848
242.3138724	291.4366019	583.9304946	510.0191257
242.3072772	291.4372059	583.9361337	510.0226017
242.3045974	291.4373906	583.9380348	510.0238953
242.3035085	291.437447	583.9386757	510.0243766

Pred values_40	Pred values_30	Pred values_20	Pred values_10
338.9028645	498.6616813	852.6322792	1235.083622
335.6753295	501.2110727	849.8797729	1244.879132
337.6979643	503.9734815	865.6952546	1254.074141
338.775126	502.9863986	866.2294884	1263.535776
337.9727642	502.5725552	879.2677932	1272.817053
337.6201851	502.8467149	882.0026586	1282.148578
337.9344174	502.88791	893.1557838	1291.411573
338.0470224	502.8239123	897.3595454	1300.667412
337.9253417	502.8258587	907.2285979	1309.884577
337.8905364	502.838893	912.4082952	1319.079448

Pred values_40	Pred values_30	Pred values_20	Pred values_10
173.3801232	234.0204673	235.7980947	299.8854714
172.5495885	230.9378699	236.9805029	299.4521528
172.692386	231.5058079	236.798621	299.560643
172.6678342	231.4011709	236.8265986	299.5334803
172.6720555	231.4204492	236.822295	299.540281
172.6713297	231.4168974	236.822957	299.5385783
172.6714545	231.4175518	236.8228552	299.5390046
172.6714331	231.4174312	236.8228709	299.5388979
172.6714368	231.4174534	236.8228684	299.5389246
172.6714361	231.4174494	236.8228688	299.5389179

Pred values_40	Pred values_30	Pred values_20	Pred values_10
297.8143411	373.728296	667.0407986	980.8433056
297.8143411	373.728296	667.0407986	980.8433056
297.8143411	373.728296	667.0407986	980.8433056
297.8143411	373.728296	667.0407986	980.8433056
297.8143411	373.728296	667.0407986	980.8433056
297.8143411	373.728296	667.0407986	980.8433056
297.8143411	373.728296	667.0407986	980.8433056
297.8143411	373.728296	667.0407986	980.8433056
297.8143411	373.728296	667.0407986	980.8433056
297.8143411	373.728296	667.0407986	980.8433056

Pred values_40	Pred values_30	Pred values_20	Pred values_10
225.7098147	211.5968701	249.2183408	263.6495224
225.3629315	213.1783501	249.3817098	263.199175
225.4231351	212.0566084	249.4110705	263.6229109
225.4126864	212.8522583	249.4163473	263.2242139
225.4144998	212.2879049	249.4172956	263.5993515
225.4141851	212.6882	249.4174661	263.2463812
225.4142397	212.4042712	249.4174967	263.5784941
225.4142302	212.6056616	249.4175022	263.2660061
225.4142319	212.4628156	249.4175032	263.5600289
225.4142316	212.5641361	249.4175034	263.2833802

Table 6.4 Predicted values for 5 banks (AXIS, HDFC, ICICI, KOTAK and SBI)

6.1.2 PROPHET

The PROPHET model had been implemented for 5 Indian banks using train dataset and least RMSE value has been calculated to find the accuracy. Train dataset has been divided in such a way (60%, 70%, 80%, and 90%) so that the best result can be evaluated. Accuracy (RMSE) has been calculated by the error between population values predicted by a model and the values observed.

Accuracy has also been calculated using 4 sets of train dataset and test dataset. Below fig has shown the RMSE value for all 5 banks for 4 sets. Least RMSE value for AXIS bank is 57.20 (90% trainset and 10% test set), HDFC bank is 465.50 (90% trainset and 10% test set), ICICI bank is 36.70 (80% trainset and 20% test set), KOTAK bank is 169.30 (90% trainset and 10% test set) and SBI bank is 27.46 (90% trainset and 10% test set).

Train	Test	RMSE Values
146 (60%)	97 (40%)	85.4952291
170 (70%)	73 (30%)	139.4265467
194 (80%)	49 (20%)	132.0441463
219 (90%)	24 (10%)	57.17100414

Train	Test	RMSE Values
167 (60%)	112 (40%)	687.9130796
195 (70%)	84 (30%)	589.8793149
223 (80%)	56 (20%)	574.7622151
251 (90%)	28 (10%)	465.5449221

Train	Test	RMSE Values
121 (60%)	80 (40%)	47.59108218
141 (70%)	60 (30%)	62.01689325
161 (80%)	40 (20%)	36.72500746
181 (90%)	20 (10%)	41.16325737

Train	Test	RMSE Values
128 (60%)	85 (40%)	449.4291813
149 (70%)	64 (30%)	448.0328095
170 (80%)	43 (20%)	176.9677967
192 (90%)	21 (10%)	169.3145037

Train	Test	RMSE Values
167 (60%)	112 (40%)	77.54350309
195 (70%)	84 (30%)	114.379111
223 (80%)	56 (20%)	43.45493515
251 (90%)	28 (10%)	27.45799787

Table 6.5 Accuracy (RMSE) values for all 5 banks (AXIS, HDFC, ICICI, KOTAK and SBI)

Below fig has shown the graphical representation of RMSE values for all 5 banks where least RMSE value can easily be identified after applying the PROPHET model.

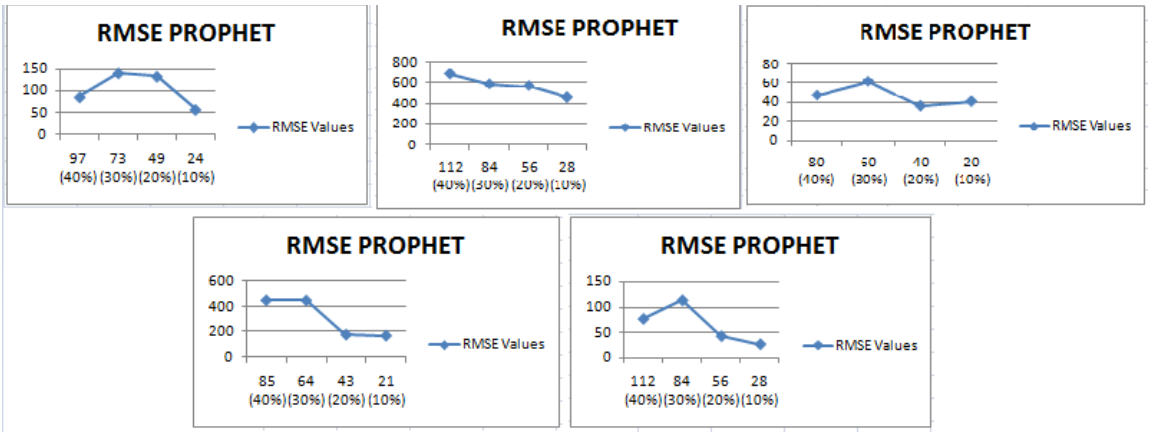


Fig 6.6 Graphical representation of RMSE values for all 5 banks (AXIS, HDFC, ICICI, KOTAK and SBI)

Below fig has shown the predicted values for all 5 banks after applying the PROPHET model for 4 trainset and test set.

Pred values_40	Pred values_30	Pred values_20	Pred values_10
237.6298468	275.7105767	341.2668104	544.2299788
245.1902297	279.5206584	351.1294229	553.8058583
253.7697587	279.7811739	357.7469229	557.1304658
255.5605151	279.1948909	361.0561604	564.0741909
260.4592625	283.7491245	365.3562378	565.788838
261.1660298	285.2513247	360.3097042	569.4520383
265.5374135	288.2537366	363.1235096	572.6965856
270.9613008	293.8515743	368.8047127	573.9355997
273.5315588	294.7924521	376.3092254	577.4671241
274.4772095	295.6082705	377.4894395	584.576187

Pred values_40	Pred values_30	Pred values_20	Pred values_10
303.4416099	505.6341314	752.2652529	1228.262476
305.6763207	512.2852887	765.4604022	1235.829718
305.5674413	509.1376079	777.6061178	1244.150633
301.881775	510.8910949	784.9609814	1256.723664
309.954032	527.5262355	791.177384	1274.302455
317.6956028	539.9712098	796.8567931	1290.758294
319.0680861	544.0232294	802.5002179	1300.051071
322.0962364	547.9209275	799.8129838	1317.389435
323.8222661	550.9290135	815.57546	1327.594496
328.8327181	550.657931	826.8138064	1346.041256

Pred values_40	Pred values_30	Pred values_20	Pred values_10
186.5555287	200.6213352	268.6983234	268.4296383
191.8145207	203.07048	271.7475586	273.6095535
192.9340387	199.672723	267.7373535	277.4520866
196.2259926	200.2341929	264.5296384	282.2596909
196.7363196	198.754645	266.6932739	283.4811739
199.0652937	204.7296216	271.4312736	285.1939039
197.7565164	209.1136636	268.347317	280.3881844
192.0522195	212.7107625	268.8649398	278.4347456
194.248151	214.2667267	268.3159096	281.6038978
195.2504858	216.7209505	272.7222451	286.5991878

Pred values_40	Pred values_30	Pred values_20	Pred values_10
229.4565463	340.6566822	619.1795875	885.5536426
236.4261916	338.5291005	632.5256569	899.1960184
243.1538442	333.7886115	646.9016548	917.3709988
242.5733559	330.4567398	665.9875381	925.3995974
248.0741818	338.9419373	676.2602187	937.0292247
254.4682929	348.110268	671.0675015	945.7140056
260.420158	348.5157573	683.4232182	951.5241235
263.8343545	352.802091	689.6220616	956.954472
265.3516437	355.5849329	704.6397618	964.5729298
270.5449241	363.8698303	706.106459	986.0956655

Pred values_40	Pred values_30	Pred values_20	Pred values_10
183.7906758	258.3470582	221.6933419	256.2675639
184.7240083	259.6161946	224.0555958	254.8185134
184.228177	257.4710928	227.0349522	249.7132482
179.8636597	258.9022873	228.5929345	249.821362
184.0083126	265.254524	228.4718926	253.0119588
190.305532	272.1775876	229.1993097	253.823065
191.8730628	275.1274019	229.6526416	254.0273005
194.1074273	275.7232166	216.5960742	255.1980608
196.0656189	277.3838812	225.0417947	254.9853993
199.2820196	276.7219518	229.9893561	258.7557955

Table 6.7 Predicted values for 5 banks (AXIS, HDFC, ICICI, KOTAK and SBI)

6.1.3 KERAS with LSTM (Long short-term memory)

The KERAS with LSTM model had been implemented for 5 Indian banks using train dataset and least RMSE value has been calculated to find the accuracy. Train dataset has been divided in such a way (60%, 70%, 80%, and 90%) so that the best result can be evaluated. Accuracy (RMSE) has been calculated by the error between population values predicted by a model and the values observed.

Accuracy has also been calculated using 4 sets of train dataset and test dataset. Below fig has shown the RMSE value for all 5 banks for 4 sets. Least RMSE value for AXIS bank is 6.31 (60% trainset and 40% test set), HDFC bank is 10.27 (80% trainset and 20% test set), ICICI bank is 3.18 (90% trainset and 10% test set), KOTAK bank is 6.42 (60% trainset and 40% test set) and SBI bank is 3.86 (70% trainset and 30% test set).

Train	Test	RMSE Values
146 (60%)	97 (40%)	6.309496112
170 (70%)	73 (30%)	7.627314237
194 (80%)	49 (20%)	12.53491778
219 (90%)	24 (10%)	8.177643154

Train	Test	RMSE Values
167 (60%)	112 (40%)	15.92855488
195 (70%)	84 (30%)	18.06708961
223 (80%)	56 (20%)	10.27311767
251 (90%)	28 (10%)	15.84698092

Train	Test	RMSE Values
121 (60%)	80 (40%)	7.020169962
141 (70%)	60 (30%)	6.423424443
161 (80%)	40 (20%)	6.561318474
181 (90%)	20 (10%)	3.184659373

Train	Test	RMSE Values
128 (60%)	85 (40%)	6.420621576
149 (70%)	64 (30%)	9.610269982
170 (80%)	43 (20%)	24.2082226
192 (90%)	21 (10%)	15.03665357

Train	Test	RMSE Values
167 (60%)	112 (40%)	7.427830613
195 (70%)	84 (30%)	3.856339641
223 (80%)	56 (20%)	6.447225014
251 (90%)	28 (10%)	8.718480211

Table 6.5 Accuracy (RMSE) values for all 5 banks (AXIS, HDFC, ICICI, KOTAK and SBI)

Below fig has shown the graphical representation of RMSE values for all 5 banks where least RMSE value can easily be identified after applying the KERAS with LSTM model.

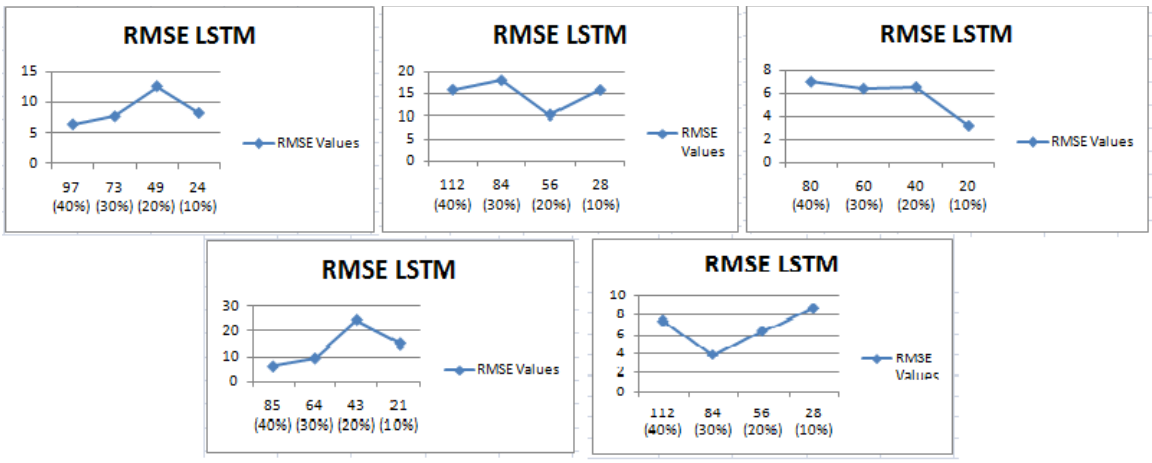


Fig 6.6 Graphical representation of RMSE values for all 5 banks (AXIS, HDFC, ICICI, KOTAK and SBI)

Below fig has shown the predicted values for all 5 banks after applying the KERAS with LSTM model for 4 trainset and test set.

Pred values_40	Pred values_30	Pred values_20	Pred values_10
247.3459959	291.1640218	596.762059	509.6231128
267.7419943	264.1245641	552.026026	507.1549421
287.941526	269.8529823	576.0533667	512.7737913
244.0223741	305.0951613	567.533313	512.1576064
253.447822	252.0501557	597.4809896	527.3018267
265.1832621	240.9404837	547.7621889	499.7612001
232.1572795	189.7457521	494.7430944	519.9194316
218.5155607	203.8292693	514.9160381	499.1905598
217.2459911	235.8550893	469.6196606	549.5442537
203.5977347	222.8528707	460.1224339	548.1717431

Pred values_40	Pred values_30	Pred values_20	Pred values_10
350.512503	545.5007045	841.0603599	1238.078878
337.3326175	511.2169712	828.1536666	1198.743804
324.7219516	545.6086351	866.8147815	1251.26882
390.0545915	601.7192506	892.6974885	1358.236113
395.211527	597.7678377	936.4415533	1430.240879
374.7543988	614.9737224	952.3935813	1490.246209
395.583619	638.8734198	1010.17171	1585.227274
412.8415354	664.788369	1073.7667	1680.452158
437.990337	697.680319	1064.401317	1718.34848
488.7889265	667.8748658	1029.070271	1784.740244

Pred values_40	Pred values_30	Pred values_20	Pred values_10
172.2569812	228.1098591	244.5165557	300.450786
172.5556725	229.3666252	235.3146274	292.753358
182.6780033	263.4659853	213.8149969	289.3548362
196.7524879	262.9955388	174.9742752	274.4158539
189.8248532	263.5100834	215.2111791	322.7541309
211.0078884	276.0798443	214.465693	306.4033569
216.4737239	282.9535903	207.5423346	335.7061434
198.1152954	274.6644329	226.9275962	322.6165164
190.9875932	322.8712769	239.3762802	290.8141191
195.909343	319.4946046	222.9846726	281.4334426

Pred values_40	Pred values_30	Pred values_20	Pred values_10
277.4263276	373.0905171	619.6083249	991.4889549
273.8926337	377.8526135	719.1613663	986.3185561
289.4641774	347.9279181	690.4493322	998.6193454
276.3169392	334.0041517	705.0549012	1019.9155
291.8493613	383.1005776	700.5394135	1067.757498
291.7854175	401.685861	634.44343	1012.458243
289.0843421	437.7464198	717.836411	1021.84949
304.5106732	456.1617165	699.3910095	1059.127251
316.7988252	455.8976211	732.0344516	1076.10968
315.9580537	499.4843716	767.6753514	1078.596421

Pred values_40	Pred values_30	Pred values_20	Pred values_10
226.5538969	219.6350271	250.790586	261.7916811
215.8436041	192.6571006	239.4374539	252.4569994
185.3179156	215.4227604	255.0185405	249.8604872
211.5118607	215.6393668	249.1119547	283.9808886
220.0083706	188.4228544	306.226961	271.3706347
231.5225348	203.6187869	312.265846	297.010892
235.5256233	229.3152348	312.7117746	291.4768516
245.3736069	210.4522074	290.7024487	281.5994262
287.1017855	235.8099551	275.8811469	291.0880923
316.8608658	253.3072216	276.7364359	284.4922833

Table 6.7 Predicted values for 5 banks (AXIS, HDFC, ICICI, KOTAK and SBI)

6.2 Comparison of Algorithms

In order to compare the results of statistical algorithms, for Axis bank ARIMA model performed well with trainset 80% and test set 20%, whereas PROPHET performed well with trainset 90% and test set 10% followed by LSTM with trainset 60% and test set 40%. Overall LSTM model performed better for all 5 banks.

Almost all 3 models performed better with larger training datasets with the exception LSTM model. ARIMA and PROPHET model performed best with 80% and 90% train dataset whereas LSTM had no trend. LSTM performed differently with each training dataset for all 5 banks.

7 CONCLUSION

This study has presented the extensive procedure of ARIMA model, PROPHET model, and KERAS with LSTM model for stock price prediction. The examinations of these 3 models uncovered that stock data set of all five banks have distinguished by each algorithm. The test results acquired with LSTM model showed the capability to predict stock prices satisfactory on a short-term basis. This could direct speculators at stock price to settle on gainful investment decisions. With the outcomes acquired LSTM model can contend sensibly well with other methods in short-term prediction.

Overall, PROPHET has the ability to work with large frequency values. The PROPHET model has been picked for a single time series which pursued some pattern for order selection. The ARIMA has a large amount of established time series techniques. The pattern segment of ARIMA was important to the point that it delivered very nearly a straight line. The LSTM prediction depended on a lot of attributes, along these lines less inclined to change because of seasonality and the current pattern. As opposed to that, the PROPHET model has worked admirably demonstrating as an added substance framework, discovering and showing seasonality. With regards to arranging future outstanding workloads, it has been observed that one could now do fine-tuning of the model. At last, keeping in mind that as of now there has been a decent pattern model which is prominent and it was not really dependably need complex machine learning calculations for determining models.

ARIMA and PROPHET generated a consistent model when there was a strong seasonal pattern in the data, whereas the LSTM model was able to identify the type of seasonality in the data and decompose the time series based on the type of seasonality and accurately predicted for all 5 banks. When comparing the statistical models and neural network, seasonal data is better handled by LSTM model than ARIMA and PROPHET, which suggests that LSTM was able to predict data with a strong pattern.

While comparing the RMSE for every model for each bank, it has been observed that the performance of statistical methods differ from Recurrent Neural Network (RNN) method because RNN method is more suitable than statistical models in predicting the stock market returns, which needs to be adapted in a case for stock price prediction.

Future Research Work

The future work of this study will be extending the number of algorithms used for stock price forecasting. The findings from this study can be used for stock price forecasting as below:

- The prediction of each bank's stock price must be analyzed to identify if there is any trend or seasonality in the data running on larger training datasets.
- Based on the trend or seasonality, a set of traditional statistical and neural network algorithm must be implemented to identify the best algorithm for stock price prediction.
- The back-testing technique must be implemented to evaluate the performance of each of the algorithm.
- The computed error terms can be expressed in RMSE for the ease of interpretation of a business user.
- The best algorithms for each stock price can be identified based on the lowest RMSE value, that algorithm must be used to predict the stock price forecasting and the successful prediction of a stock price might end up with significant profit.

8 PLAGIARISM AND REFERENCES

Brockwell, P.J., Davis, R.A. and Calder, M.V., 2002. Introduction to time series and forecasting (Vol. 2). New York: springer.

Granger, C.W.J. and Newbold, P., 2014. Forecasting economic time series. Academic Press.

Akhilesh Ganti, Mar, 2019. www.investopedia.com/terms/s/stock.asp

A. Victor Devadoss and T. Antony Alphonnse Ligori, 2013. Forecasting of Stock Prices Using Multi Layer Perceptron.

X. Ding, Y. Zhang, T. Liu, and J. Duan, 2015. Deep learning for event-driven stock prediction.

Menon, V.K., Vasireddy, N.C., Jami, S.A., Pedamallu, V.T.N., Sureshkumar, V. and Soman, K.P., 2016, June. Bulk price forecasting using spark over NSE data set. In International Conference on Data Mining and Big Data (pp. 137-146). Springer, Cham.

Hiransha, M., Gopalakrishnan, E.A., Menon, V.K. and Soman, K.P., 2018. NSE stock market prediction using deep-learning models. Procedia computer science, 132, pp.1351-1362.

Ariyo, A.A., Adewumi, A.O. and Ayo, C.K., 2014, March. Stock price prediction using the ARIMA model. In 2014 UKSim-AMSS 16th International Conference on Computer Modelling and Simulation (pp. 106-112). IEEE.

Pai, P.F. and Lin, C.S., 2005. A hybrid ARIMA and support vector machines model in stock price forecasting. Omega, 33(6), pp.497-505.

Wirth, R. and Hipp, J., 2000, April. CRISP-DM: Towards a standard process model for data mining. In Proceedings of the 4th international conference on the practical applications of knowledge discovery and data mining (pp. 29-39). Citeseer.

Zazzaro, Gaetano & Romano, Gianpaolo & Mercogliano, Paola. (2017). Data Mining for Forecasting Fog Events and Comparing Geographical Sites. Designing a novel method for predictive models portability. International Journal on Advances in Networks and Services. 10. 160-171.

Gardner, M.W. and Dorling, S.R., 1998. Artificial neural networks (the multilayer perceptron)—a review of applications in the atmospheric sciences. Atmospheric environment, 32(14-15), pp.2627-2636.

Sean J. Taylor & Benjamin Letham (2018) Forecasting at Scale, The American Statistician, 72:1, 37-45, DOI: 10.1080/00031305.2017.1380080

Mingyue, Q., Cheng, L. and Yu, S., 2016, July. Application of the Artifical Neural Network in predicting the direction of stock market index. In 2016 10th International Conference on Complex, Intelligent, and Software Intensive Systems (CISIS)(pp. 219-223). IEEE.

Montgomery, D.C., Jennings, C.L. and Kulahci, M., 2015. Introduction to time series analysis and forecasting. John Wiley & Sons.

Lo, A.W. and Wang, J., 2001. Stock market trading volume.

Gulli, A. and Pal, S., 2017. Deep Learning with Keras. Packt Publishing Ltd.

Devi, B.U., Sundar, D. and Alli, P., 2013. An effective time series analysis for stock trend prediction using ARIMA model for nifty midcap-50. International Journal of Data Mining & Knowledge Management Process, 3(1), p.65.

C Olah, C., 2015. Understanding lstm networks.

Taylor SJ, Letham B. 2017. Forecasting at scale. PeerJ Preprints 5:e3190v2 https://doi.org/10.7287/peerj.preprints.3190v2

Choi, K., Joo, D. and Kim, J., 2017. Kapre: On-gpu audio preprocessing layers for a quick implementation of deep neural network models with keras. arXiv preprint arXiv:1706.05781.

9 APPENDICES

This document will guide you through the contents of the Artifacts and the necessary steps to implement the R code for dissertation project titled "STOCK PRICE PREDICTION USING TIME SERIES MODELS".

9.1 Contents of the Artifacts

9.1.1 Datasets

- DataPreparation.R: This R code pre-process the data for the requirement before modeling
- AXISBANK.csv: The pre-processed data file for AXIS bank.
- HDFCBANK.csv: The pre-processed data file for HDFC bank.
- ICICIBANK.csv: The pre-processed data file for ICICI bank.
- KOTAKBANK.csv: The pre-processed data file for KOTAK bank.
- SBIBANK.csv: The pre-processed data file for SBI bank.
- Axis_Monthly_Avg_Share.csv: Monthly average data file for AXIS bank.
- HDFC_Monthly_Avg_Share.csv: Monthly average data file for HDFC bank.
- ICICI_Monthly_Avg_Share.csv: Monthly average data file for ICICI bank.
- Kotak_Monthly_Avg_Share.csv: Monthly average data file for KOTAK bank.
- SBI_Monthly_Avg_Share.csv: Monthly average data file for SBI bank.

9.2 Model Execution

9.2.1 ARIMA

- Arima_AXIS.R: This R code split the data into Trainset and test set and ARIMA model execution for AXIS bank.
- Arima_HDFC.R: This R code split the data into Trainset and test set and ARIMA model execution for HDFC bank.
- Arima_ICICI.R: This R code split the data into Trainset and test set and ARIMA model execution for ICICI bank.
- Arima_KOTAK.R: This R code split the data into Trainset and test set and ARIMA model execution for KOTAK bank.
- Arima_SBI.R: This R code split the data into Trainset and test set and ARIMA model execution for SBI bank.

9.2.2 PROPHET

- Prophet_AXIS.R: This R code split the data into Trainset and test set and PROPHET model execution for AXIS bank.
- Prophet_HDFC.R: This R code split the data into Trainset and test set and PROPHET model execution for HDFC bank.
- Prophet_ICICI.R: This R code split the data into Trainset and test set and PROPHET model execution for ICICI bank.
- Prophet_KOTAK.R: This R code split the data into Trainset and test set and PROPHET model execution for KOTAK bank.
- Prophet_SBI.R: This R code split the data into Trainset and test set and PROPHET model execution for SBI bank.

9.2.3 KERAS with LSTM

- LSTM_AXIS.R: This R code split the data into Trainset and test set and KERAS with LSTM model execution for AXIS bank.
- LSTM_HDFC.R: This R code split the data into Trainset and test set and KERAS with LSTM model execution for HDFC bank.
- LSTM_ICICI.R: This R code split the data into Trainset and test set and KERAS with LSTM model execution for ICICI bank.
- LSTM_KOTAK.R: This R code split the data into Trainset and test set and KERAS with LSTM model execution for KOTAK bank.
- LSTM_SBI.R: This R code split the data into Trainset and test set and KERAS with LSTM model execution for SBI bank.

9.2.4 Model Result

- graphs.R: This R file contains R code for graphs including all 3 models for all 5 banks.
- Results_ARIMA.xlsx: This file contains result for all 5 banks including AIC values, RMSE values, trainset and test set % vise data count, and predicted values.
- Results_Prophet.xlsx: This file contains result for all 5 banks including RMSE values; trainset and test set % vise data count, and predicted values.
- Results_LSTM.xlsx: This file contains result for all 5 banks including RMSE values; trainset and test set % vise data count, and predicted values.
- Combined result RMSE.xlsx: This file contains combined result for RMSE values for all 5 banks.