**COMPARATIVE STUDY OF TIME SERIES PREDICTION MODELS IN CONTEXT OF INDIAN STOCK MARKET**

**AMIT CHAWLA**

**Mid Thesis Report**

**DECEMBER 2020**

# Abstract

With the ever-decreasing bank interest rates, increasing investor awareness and risk appetite, the stock market is becoming a preferred investment tool for a common man. However, in addition to investments, short-term stock trading is an important part of financial markets. Most of the traders use leverage to make short-term trades. Due to leverage, the accuracy of the trading decisions is key to success. Over time, traders have developed various technical indicators to aid financial decisions. Stock market data is essentially a time series data that exhibits characteristics random walk. Researchers have attempted to maximize stock market prediction accuracy using machine learning and deep learning algorithms as computational resources have become more widely available. However, like in many business problems, forecasting at scale has been a challenge for machine learning algorithms. Interpretability of the deep learning methods has been another key challenge area for the researchers in the case of stock market prediction. This study focuses on evaluating novel time series forecasting methods against the state-of-art time series prediction algorithms. The study studies the effect of popular technical indicators on these models. The study also enhances the existing literature by assessing the performance of the state-of-the-art machine learning algorithms on the Indian stock market for the years 2020-2021. This research will provide analysts and traders with new methods for making predictions on the Indian stock market based on long term dependencies and reduce bottlenecks involved in forecasting at scale.

# Table of contents

[Abstract 2](#_Toc75048997)

[Table of contents 3](#_Toc75048998)

[LIST OF fIGURES 5](#_Toc75048999)

[LIST OF TABLES 6](#_Toc75049000)

[LIST OF ABBREVIATIONS 7](#_Toc75049001)

[Chapter 1 Introduction 9](#_Toc75049002)

[1.1 Background Of the Study 9](#_Toc75049003)

[1.2 Problem Statement 10](#_Toc75049004)

[1.3 Aim and Objectives 11](#_Toc75049005)

[1.4 Research Questions 11](#_Toc75049006)

[1.5 Scope of the Study 11](#_Toc75049007)

[1.6 Significance of the Study 12](#_Toc75049008)

[1.7 Structure of the Study 13](#_Toc75049009)

[Chapter 2 Literature Review 14](#_Toc75049010)

[2.1 Time Series Analysis 14](#_Toc75049011)

[2.2 Studies based on Technical Indicators 21](#_Toc75049012)

[2.3 Studies based on Sentiment Analysis for stock market. 29](#_Toc75049013)

[2.4 Studies based on Time Series Similarity 32](#_Toc75049014)

[2.5 Summary 34](#_Toc75049015)

[Chapter 3 Research Methodology 36](#_Toc75049016)

[3.1 Introduction 36](#_Toc75049017)

[3.2 Data Collection 36](#_Toc75049018)

[3.3 Data Profiling 36](#_Toc75049019)

[3.4 Initial Exploratory Data Analysis 38](#_Toc75049020)

[3.5 Train and Test Split 39](#_Toc75049021)

[3.6 Data Cleaning and Feature Engineering 39](#_Toc75049022)

[3.6.1 Data Cleaning 39](#_Toc75049023)

[3.6.1 Feature Engineering 39](#_Toc75049024)

[3.6.2 Data pre-processing and transformations 41](#_Toc75049025)

[3.7 Detailed Exploratory Data Analysis 41](#_Toc75049026)

[3.8 Model Building 41](#_Toc75049027)

[3.8.1 Auto Regression 42](#_Toc75049028)

[3.8.2 Moving Average (MA) 42](#_Toc75049029)

[3.8.3 Trend Differencing(I) 42](#_Toc75049030)

[3.8.4 ARMAX 42](#_Toc75049031)

[3.8.5 Seasonal Autoregressive Integrated Moving Average Exogenous model (SARIMAX) 43](#_Toc75049032)

[3.8.6 Vector Autoregressive Moving-Average Processes with Exogenous Regressors (VARMAX) 43](#_Toc75049033)

[3.8.7 Prophet 44](#_Toc75049034)

[3.8.1 AR-Net - Auto-Regressive Neural Network for time-series 45](#_Toc75049035)

[3.9 Model Validation 47](#_Toc75049036)

[3.9.1 Mean Absolute Percentage Error (MAPE) 47](#_Toc75049037)

[3.9.2 Root mean squared error (RMSE) 48](#_Toc75049038)

[3.9.3 Adjusted R-squared 48](#_Toc75049039)

[3.9.4 Noise 49](#_Toc75049040)

[3.10 Comparison of the results 49](#_Toc75049041)

[3.11 Model Interpretability 49](#_Toc75049042)

[3.12 Summary 51](#_Toc75049043)

[References 52](#_Toc75049044)

[Appendix A Research Plan 72](#_Toc75049045)

[Appendix B Research Proposal 73](#_Toc75049046)

# LIST OF fIGURES

[Figure 2.1 Volatility clustering (Chang and Tsai, 2008) 14](#_Toc75051194)

[Figure 2.2 Overshoot problem (Chang and Tsai, 2008) 14](#_Toc75051195)

[Figure 2.3 Uncertainty-aware Attention (UA) model structure (Gao et al., 2020) 22](https://nagarro-my.sharepoint.com/personal/amit_chawla02_nagarro_com/Documents/Projects/DataScience/Thesis%20work/Chapters/Mid%20Thesis%20Amit%20Chawla.docx#_Toc75051196)

[Figure 2.4 Technical indicators (Nabipour et al., 2020) 24](https://nagarro-my.sharepoint.com/personal/amit_chawla02_nagarro_com/Documents/Projects/DataScience/Thesis%20work/Chapters/Mid%20Thesis%20Amit%20Chawla.docx#_Toc75051197)

[Figure 2.5 An AR equivalent architecture. (Triebe et al., 2019) 28](https://nagarro-my.sharepoint.com/personal/amit_chawla02_nagarro_com/Documents/Projects/DataScience/Thesis%20work/Chapters/Mid%20Thesis%20Amit%20Chawla.docx#_Toc75051198)

[Figure 2.6 An AR inspired neural network with multiple layers. (Triebe et al., 2019) 28](https://nagarro-my.sharepoint.com/personal/amit_chawla02_nagarro_com/Documents/Projects/DataScience/Thesis%20work/Chapters/Mid%20Thesis%20Amit%20Chawla.docx#_Toc75051199)

[Figure 2.7 Prediction Model for Daily Prediction Model (Nayak et al., 2016) 30](#_Toc75051200)

[Figure 2.8 A configuration tree of all the experiments performed by (Sidi, 2020) 33](#_Toc75051201)

[Figure 3.1 Research Methodology 37](https://nagarro-my.sharepoint.com/personal/amit_chawla02_nagarro_com/Documents/Projects/DataScience/Thesis%20work/Chapters/Mid%20Thesis%20Amit%20Chawla.docx#_Toc75051202)

# LIST OF TABLES

[Table 2.1 Results summary (Ji et al., 2019) 17](#_Toc75051228)

[Table 2.2 Metric for UA attention models (Gao et al., 2020) 22](#_Toc75051229)

[Table 2.3 Experiments and Results for the combination of Facebook Prophet and Atten-LSTM (Nagesh, 2021) 27](#_Toc75051230)

[Table 2.4 Accuracy of the daily prediction model (Nayak et al., 2016) 31](#_Toc75051231)

[Table 3.1 Features of the dataset. 38](#_Toc75051232)

[Table 3.2 Technical Indicators 40](#_Toc75051233)

[Table 3.3 Pros and Cons of Models. 46](#_Toc75051234)

[Table 3.4 Pros and Cons of Evaluation Metrics 48](#_Toc75051235)

[Table 3.5 Model Interpretability Tools 50](#_Toc75051236)

# LIST OF ABBREVIATIONS

| **Abbreviation** | **Expansion** |
| --- | --- |
| ADX | Average Directional Movement Index |
| AMH | Adaptive Market Hypothesis |
| ANN | Artificial Neural Network |
| API | Application Programming Interface |
| AR | Auto Regressive |
| ARCH | Auto-Regressive Conditional Heteroscedasticity |
| ARIMA | Autoregressive Integrated Moving Average |
| ARMA | Autoregressive Moving Average |
| ARMAX | Autoregressive Moving Average with Exogenous Variables |
| ATR | Average True Range |
| BPNN | Back Propagation Neural Networks |
| BSE | Bombay Stock Exchange |
| CAGR | Compound Annual Growth Rate |
| CCI | Commodity Channel Index |
| CNN | Convolutional Neural Network |
| DJI | Dow Jones Industrial Average |
| DTW | Dynamic Time Warping |
| EMA | Exponential Moving Average |
| EMH | Efficient Market Hypothesis |
| FIS | Fuzzy Inference System |
| GARCH | Generalized Autoregressive Conditional Heteroskedasticity |
| HAN | Hybrid Attention Network |
| HDFS | Hadoop Distributed File System |
| LIME | Local Interpretable Model-Agnostic Explanations |
| LSTM | Long Short-Term Memory |
| MA | Moving Average |
| MACD | Moving Average Convergence Divergence |
| MAE | Mean Absolute Error |
| MAPE | Mean Absolute Percentage Error |
| MCC | Matthews Correlation Coefficient |
| MLP | Multilayer Perceptron |
| MSE | Mean Square Error |
| NDN | News Distilling Network |
| NSE | National Stock Exchange |
| OBV | On Balance Volume |
| OHLCV | Open High Low Close Volume |
| PCA | Principal Component Analysis |
| PROC | Predicting The Rate Of Change Price |
| RBF | [Radial Basis Function](https://towardsdatascience.com/radial-basis-function-rbf-kernel-the-go-to-kernel-acf0d22c798a) |
| RF | Random Forest |
| RMSE | Root Mean Squared Error |
| RNN | Recurrent Neural Network |
| ROC | Rate Of Change |
| RSI | Relative Strength Index |
| SANN | Salience-Affected Neural Network |
| SARIMAX | Seasonal Autoregressive Integrated Moving Average Exogenous Model |
| SAX | Symbolic Aggregate Approximation |
| SHAP | Shapley Additive Explanations |
| SMA | Simple Moving Average |
| STAR | Autoregressive Soft Transition |
| STD | Standard Deviation |
| SVM | Support Vector Machine |
| SVR | Support Vector Regression |
| TAR | Autoregressive Thresholds |
| TEMA | Triple Exponential Moving Average |
| TRIX | Triple Moving Exponential |
| UA | Uncertainty Aware Attention |
| VARMAX | Vector Autoregressive Moving-Average Processes with Exogenous Regressors |
| VIF | Variance Inflation Factor |
| WMA | Weighted Moving Average |
| ADF | Augmented Dickey Fuller |

# Introduction

## Background Of the Study

The stock market is an important component of a nation’s economy. Research has shown that stock market capitalization positively impacts the economic growth of a country(Aali-Bujari et al., 2014). Any major political, economic, even natural developments that impact the economic growth of the country also directly impact the stock market movement and vice versa. Over the last 3 decades, SENSEX has delivered approximately 11.74% of CAGR returns. The stock market has been one of the most rewarding yet among the most feared investment avenue. However, the recent break of the COVID-19 pandemic that led to nationwide lockdown led to a crash in the market wiping out crores of rupees worth of investor wealth. On the other hand, the sharp recovery of the markets provided ample opportunity for the investors to deploy any idle cash for superior returns. Another segment of people who benefit from the rewarding nature of the stock market is the traders. Traders usually buy/sell leveraged positions in the market for a short duration, ranging from few seconds to a few days. There are roughly 4500 stocks listed on the Bombay stock exchange. Since most of the traders take leveraged positions, they prefer to do so in Futures or liquid scripts only to facilitate higher margins and easy square-off. However, prediction of the stock markets has been a challenging task so much so that (Fama, 1995) proposed an efficient market hypothesis (EMH) stating that stock market prices already factor in all the available information, and it is not possible to predict future prices. EMH challenges the rationality of the markets. (Lo, 2004) proposes adaptive market hypothesis (AMH) and suggested tacking the stock market from a biological perspective. It attempts to overcome the criticism of EMH by stating that stock prices evolve according to competition, adaption, and natural selection to financial interactions. It states that predictable patterns for stock prices may appear in short term. This is an ongoing debate. The increasing availability of data and computing resources has encouraged research to improve the accuracy of trading systems. However, as discussed in Chapter 2, the focus of the research has been on short term forecasts and for developed markets. This research aims to address these gaps and study the predictive power of machine learning algorithms in the context of the Indian stock market and for long-range dependencies.

## Problem Statement

The fundamental purpose of time series modelling is to identify a global framework, mathematical formulas to describe the behaviour of the data generating process and forecast future behaviour by knowing the past. Popular theories suggest that stock markets are essentially a random walk. Indeed, the stock values are generally dynamic, non-parametric, and non-linear. (Nabipour et al., 2020). The research for stock market time series which is based on technical theories has a basic assumption that history repeats itself (Nayak et al., 2016). Various studies have tried to break the stock market data into seasonality and trend and noise components to study the behaviour of time series and make a prediction. Neural networks have proven to be efficient for stock market data, but the high number of factors and configurations make them unstable technique. Also, these techniques are not explainable. (Andrea Sánchez-Sánchez et al., 2020). A recent evaluation of the machine learning models by (Gerlein et al., 2016) finds that with an optimal combination of training set size, attribute set and periodic retraining, machine learning models can achieve performance like more complex counterparts. At the same time some studies have shown that for simple machine learning models for time series prediction, the autocorrelation decays exponentially as the time gap increases. This study investigates the viability of employing classic machine learning models for long-term dependencies in the stock market by simulating the standard AR model using neural networks. Since, simple regression cannot be applied to estimate the regression coefficients due to the non-stationarity of data, this study proposes to employ neural networks for the same. The study proposes to use AR-Net, a novel method that scales for long-range dependencies while tending to keep the model information explainable and computationally feasible. This study investigates the efficiency of a novel time-series prediction model called the Prophet for stock market prediction. The Prophet is defined as a linear combination of trend, seasonality and holidays or special events. These are key characteristics of any business time series including the stock market. Prophet has been designed for forecasting at scale while keeping the model interpretable and configurable. The literature review reveals that no study for benchmarking the performance of traditional time series forecasting models and the novel time-series prediction models - Prophet (Taylor and Letham, 2017), and Auto-Regressive Neural Network (Triebe et al., 2019) for the Indian stock market has been done. Since times 2020 and 2021 are unprecedented, this study will also evaluate the performance of the state-of-art time series models for the stock market data including the above-mentioned period and compare the performance of these models with Prophet and Auto-Regressive Neural Network.

## Aim and Objectives

The main aim of this research is to propose a prediction model for stock market prediction. The primary goal of this research is to explore and demonstrate the performance of various state-of-art and novel time series prediction algorithms in context to the stock market data.

The research objectives formulated based on the aim of this study are as follows:

* To analyse the use of technical indicators with the price movement for finding and analysing trade patterns.
* To compare the performance of Prophet and AR-Net model with conventional time-series prediction models.
* To analyse the performance of proposed models during the COVID period.
* To propose an optimal combination of model and technical indicators for stock market prediction.

## Research Questions

Literature review and the identified gaps lead to the following research questions:

* Can the derived technical variables enhance the performance of classical time series algorithms?
* Are the novel algorithms comparable to the state-of-the-art time series prediction algorithms?
* How does the use of a neural network for determining AR coefficients impact the values of the coefficients?
* Are the newly calculated AR coefficients able to explain the behaviour and nature of stock market data?
* Is the novel neural network framework interpretable yet effective for the stock market prediction?

## Scope of the Study

The scope of this study is limited to the following points:

* This study will use daily frequency data for fifteen stocks and sector indices for five different sectors.
* Data is based on trades done on the Bombay stock exchange.
* This study is limited to the technical analysis of the stock.
* This study makes use of up to 6 derived technical indicators as external variables for model building in addition to features available with the dataset.
* The impact of data transformation techniques is only investigated without the exogenous variables.
* The study will compare the performance of different models for the given dataset.

This study does not cover the following:

* Trade data from National Stock Exchange (NSE) are not part of the dataset feature calculations. This is due to the unavailability of open-source data from NSE. All the data sources considered during the proposal either charge heavily or do not provide enough data in a free trial.
* This study is based on the technical and statistical analysis of the time-series data. The objective of the study is to model the autocorrelation of the data and the statistically derived indicators for stock market prediction. Therefore, financial information for the company is not provided as input to the models.
* Due to the above-mentioned reasons, no macro-economic events or factors are taken into consideration.
* The objective of the study is to analyse the performance of novel algorithms in modelling higher-order lags or dependencies of the time-series data. Since this is based on mathematical relationships, news or special events specific to the stock are not included in the input parameters.

## Significance of the Study

Stock market forecasting has always been a difficult job to accomplish. The random walk aspect of the stock market time series, as well as the large number of variables involved, are the primary reasons for this. However, studies have shown that historical data reflects stock prices, making movement patterns relevant for forecasting (Akhter and Mısır, 2005). Multiple studies have suggested that classical machine learning models either become too slow or too inefficient for long-range dependencies. Although neural network models like RNN, LSTMs have known to be more effective in the context of the stock market, their results are not interpretable. The study is significant because it provides a simple, configurable, and yet effective model for stock market prediction. This will enable traders with no coding or technical background to build trading systems. The models testing on data for the COVID era ensures the performance of the models in volatile markets hence providing a reliable method for any market conditions. This study will also contribute to the existing literature by providing a benchmark for modelling linear time series models for non-stationary and dynamic time-series using neural networks.

## Structure of the Study

The study is divided into the following subsections-

Chapter 1 discussed briefly the various theories proposed in the context of the stock market and highlights the challenges faced by investors and traders in predicting the stock market. It discusses the key aims and objectives of the study. The chapter then raises key research questions, answers to which are sought through this study. The scope of the study and its significance is discussed in this chapter.

Chapter 2 discusses various approaches proposed by the researchers for stock market predictions. It also reviews various novel methods proposed for stock market or time-series prediction in general. It discusses the various gaps in previous research and sets the context for this research.

Chapter 3 discusses the approach taken to achieve the aims and objectives set for this study. This chapter explains the data sources and data collection methods relevant to this study. It discusses the need for pre-processing of data and steps taken for the same. It also mentions the context for the exploratory data analysis in this study. The formulations for various technical indicators used in this study are discussed in subsection 3.6.1. It briefly discusses the idea behind various algorithms and their applicability in the current context. Further, it explains the evaluation metrics and their relevance for this study. It also presents an argument for using relevant methods for interpreting the models.

# Literature Review

This section examines the previous literature and, where applicable, provides a critical review. The performance of various machine learning, deep learning, and novel algorithms is reviewed in this section. It also presents the various challenges faced in time-series and stock market predictions in different contexts. Additionally, this part clarifies the gaps in the existing research and formulates the study's aim.

## Studies based on time series analysis

The main goal of time series modelling is to find a global structure, mathematical formulas to describe the behaviour of the data generating process and predict the future behaviour through the understanding of the past. Financial time series are among the complex time series due to a low signal-to-noise ratio (LALOUX et al., 2000) and heavy-tailed distribution (Cont, 2001). Also, the relation between past and future values is not deterministic. This makes it very tough to predict the values of stock price indexes.

In complex time series, the predictive accuracy of various models is impacted by two major problems- the overshoot phenomenon, and volatility clustering. Classic time series prediction algorithms like ARMA do not consider the aspect of time-varying conditional variance in the residuals also referred to as volatility clustering. When performing prediction on complex time series like financial data but with a small amount of data, the traditional models like the grey model, lead to high residual errors. This is also referred to as the overshooting problem.

|  |  |
| --- | --- |
| Chart  Description automatically generated | A picture containing diagram  Description automatically generated |
| Figure 2.1 Volatility clustering (Chang and Tsai, 2008) | Figure 2.2 Overshoot problem (Chang and Tsai, 2008) |

Figure 2.1 represents shows the volatility clustering and Figure 2.2 depicts the overshoot problem. Further, the non-linear models have challenges of their own like – the complexity of the function for non-linear models, the evaluation of model parameters, and the kind of model itself. (Chang and Tsai, 2008) propose to divide and conquer the above two challenges in a complex time series forecast. Therefore, it proposes an ensemble of Grey Model tuned with adaptive support vector regression also referred to as SVRGM and GARCH models through a backpropagation neural network to achieve higher accuracy on the non-periodic short-term forecast. GARCH helps in dealing with volatility clustering. SVRGM, enabled by its self-adaptive learning ability, fits well on irregular or non-periodic time series for short-term predictions. The composite model is a linear combination of SVRGM and GARCH given by the equation (2.1).

|  |  |  |
| --- | --- | --- |
|  |  | (2.1) |

To find the optimal values of the parameters for the above relation, linear regression is not suited as it cannot capture the correct behaviour of non-stationary data due to highly nonlinear dynamics. Therefore, a standard three-layer back-propagation neural network is used to find optimally approximate values of coefficients. The study uses a single-step-look-ahead prediction methodology. A small number of the most recent observed data is collected as a data queue for modelling an intermediate predictor to predict the next cycle performance in single-step-look-ahead architecture. Once the sampled datum for the next time is collected, a datum at the bottom of the data queue is removed and the most recent sampled datum is added at the top, creating the current data queue used for the next prediction. The experiment has been designed to evaluate the performance of the proposed model against various benchmark models like the grey model, ARMA, Holt winter’s smoothing, radial basis function neural networks. The performance is evaluated on monthly closing prices of four key indices of the stock market – Dow Jones Industrial Index, London FT-100 Index, Tokyo Nikkei Index, Taipei TAIEX index, futures, and options index provided by London International Financial Futures and Options Exchange, for 36 months. The models are evaluated on mean absolute deviation, MAPE, and MSE. The study observes that ARMA is the worst affected due to volatility clustering. Grey Model has high residual errors for areas of turning points like trend change etc. The proposed ensemble of SVRGM/GARCH when tuned through Backpropagation neural networks (BPNN) achieves the task of improving prediction accuracy for non-periodic short-term forecasts. However, an ideal definition for a short-term period in the context of traders can vary from few seconds to a couple of days. The study has not considered intraday, daily, weekly data for short-term forecasts.

Short-term fluctuations tend to test investor's patience, however, historically the stock markets have outperformed other asset classes. Investors a technique called trend following for making trading decisions and overcome the effect of short-term volatility and fluctuations. (Abraham et al., 2001) attempts to understand the insights derived from group trend analysis. The literature review has revealed that neural networks have mostly outperformed other methods for time-series predictions. Fuzzy inference systems (FIS) have excellent decision-making capability under uncertainty. (Abraham et al., 2001) proposes to use neural network training algorithms to fine-tune fuzzy inference systems. The study uses data of six industry-leading stocks and the Nasdaq-100 index for two years. The authors have used principal component analysis (PCA) for feature extraction. The scaled Conjugate Gradient algorithm (Møller, 1993) is used to train the ANN in over 2000 epochs. This study employs a neuro-fuzzy system for trend analysis i.e., to determine the direction of the movement of the stock. A five-layer EFuNN is used as FIS in this study. The second layer of nodes represents fuzzy quantification of each input variable space after the input layer. Each input variable is expressed by a cluster of spatially organized neurons, which represents a fuzzy quantization of the variable. The authors also assert that different membership functions for each neuron and can be modified during learning. A rule-based third layer evolves through learning. Fuzzy quantification of the input problem space is provided by the fourth layer and used by the fifth to represent the actual value.

Experiments show that the testing error obtained by the algorithm is maximum for Intel and Microsoft @0.034 and lowest for yahoo @ 0.021. The EFuNN clocks a success rate of 100%. The study only uses two years of data. This amount of data along with a high number of epochs can lead to the overfitting of neural networks. This also makes the process computationally expensive. The success rate may not be generalizable over the longer training and testing periods. The study has not considered validating the results for a larger training dataset or a longer prediction duration. Due to a lack of data, the influence of numerous market occurrences has not been examined.

Just like the stock market, bitcoin has become an important financial asset. Bitcoin prices also act as a random walk. (Ji et al., 2019) compares the performance of state-of-art and novel neural network-based models for bitcoin price prediction.

Table 2.1 Results summary (Ji et al., 2019)

|  |  |  |
| --- | --- | --- |
| **Parameter under observation** | **Regression** | **Classification** |
| Sequence size(m) | m = 5 – DNN  m = 10,20 - LSTM  m= 50, 100 – SVM | For the same values of m, there is no clear winner. |
| Effect of using Log values (m = 20) | Accuracy of all the models except ResNet and CRNN increases. | For CRNN and SVM model accuracy does not improve with log transformation. DNN shows a major improvement in accuracy. |
| Effect of data split methods- Sequential, Random split, 5-fold CV | (m = 20)  LSTM performs best for all 3 kinds of splits. All models log the best performance for the random split. | (m = 50)  DNN performs the best for all 3 kinds of splits. 5-fold CV is however the best-suited split for DNN. Different models perform best with different kinds of splits. There is no clear winner. |
| Effect of Normalization Methods – first value, min-max (m=20) | LSTM performs best for both normalization methods, however, there is a significant improvement in MAPE value for first value normalization | DNN performs best for both normalization techniques. However, the improvement in accuracy is not very significant. There is no clear winner that fits all models well. |

The study proposes to include a deep residual network that has proven effective for long sequence data analysis (Dauphin et al., 2016). The study proposes to develop both regression and classification problems for the bitcoin price prediction problem. The authors have obtained data for seven years of bitcoin price movement along with 29 other features. This study uses Spearman rank correlation coefficients (Spearman, 1987) for selecting eighteen features with a correlation value to bitcoin price between 0.75 and 0.95. This study uses log transformation for scaling. The study also confirms that the first value-based normalization is more effective than min-max normalization (Ji et al., 2019). The study uses SVM with RBF kernel as the baseline method. The study uses MAPE, accuracy as statistical measures for regression and classification, respectively. The key findings of this study are enlisted in Table 2.1. For different sequence sizes, the study observes that for regression, LSTMs perform well for smaller sequence sizes while SVMs outperform other models by a large margin for sequence sizes exceeding fifty. In addition to being non-stationary like stock prices, the price movement of cryptocurrencies is extremely volatile. (Mudassir et al., 2020) proposes to build machine learning models to predict prices for bitcoin for longer time horizons. This study divides the data into three different intervals that represent three different aspects – low volatility and volumes, high valuation period and, high volatility interval. (Mudassir et al., 2020) propose to use SMA, EMA, RSI, STD, WMA, TRIX, ROC as the technical indicators. Authors have used Random Forest, VIF, and Pearson correlation as metrics to perform feature selection. This study builds both classification and regression models for predicting the direction and the absolute price of the bitcoin for a 7, 30, 90-day time horizon. Experiments conducted by (Mudassir et al., 2020) show that, among regression models, SANN outperforms others for all the time horizons in low volatility intervals. The study also finds SVMs to be more effective in high volatility settings. LSTMs not only outperforms the peers for classification task but have also proven to be the most stable for both regression and classification. The authors also conclude that PCA helps in improving classification accuracy significantly.

(Lam, n.d.) discusses multiple business use cases for time series analysis and the challenges associated with them. A time series is essentially a set of observations made at various points in time. Time-series observations have a natural ordering as time moves forward. This separates time-series data from other types of data since data points that are "near" in time are expected to have more in common than data points that are "further" apart in time.

Analysis of time series is essentially considered to be a study of autocorrelations in data. The definition of the AR (1) model implies the dependency of values on immediate past values. It can further be interpreted that autocorrelation decays exponentially as the time gap increases (Lam, n.d.). The goal of time series analysis is to have a simple and explainable model that can explain the autocorrelation of data and provide useful forecasts. As demonstrated by the study of climate data with 1540 components for 528 length time series, a vector autoregressive model can lead to over 23.7 million parameters to be estimated. This leads to a huge least square problem. Adding additional variables like air pressure, rainfall etc makes the task more complicated.

Similar issues occur for time series with high noise. The algorithms should be able to distinguish between signals and background noise. In finance, understanding the relationship between different kinds of financial parameters can be reduced to a problem of estimation of sparse precision matrix. A zero-entry means independent variables. For determining the asset allocation of the portfolio, the precision matrix of a set of stock returns is analysed to ensure diversification. Any change in the network or matrix means a change in optimal allocation. To add to it, the detection of change in the network of stocks poses another challenge. Timely and accurate detection is important for portfolio managers to be able to invest for decent returns. The increasing availability of computing resources has enabled the analysis and extraction of information, but it still relies on the innovations of statisticians.

Most business time series have certain common characteristics- namely seasonality, trend, outliers, and holiday effects. There are certain challenges in business time series forecasting. Automatic forecasting methods can be difficult to fine-tune and are often too rigid to incorporate useful assumptions or heuristics. Also, analysts need to be well-trained in forecasting techniques in addition to domain expertise. This makes forecasting of time series at scale, a challenging task. As it has been observed in the preceding literature review, the traditional methods like Auto ARIMA fail to capture seasonality, exponential smoothing and seasonal naïve forecasts struggle to capture long-term seasonality in time series data. Neural networks and other novel mechanisms based on neural networks are either not scalable or too expensive computationally. (Taylor and Letham, 2017) address three aspects of scale through a novel algorithm - many people making forecasts without specific training on underlying models, a wide range of forecasting issues with potentially unique characteristics, and many forecasts thus needing an automated evaluation mechanism. This study proposes a modular regression model with observable parameters. A decomposable time series model (Harvey and Peters, 1990) with 3 main components – trend, seasonality, and holidays, is used by the study. The equation for the model is shown in equation (2.2).

|  |  |  |
| --- | --- | --- |
|  |  | (2.2) |

represents the trend, represents seasonality, represents holidays and represents irregular changes and is assumed to have a normal distribution. The model essentially reduces a time series forecasting problem to a curve-fitting task. Two trend models have been implemented – a nonlinear model to model the saturating growth and a linear trend model with changepoints. The nonlinear model is represented as a logistic regression model as shown in the equation below and allows to account for variable growth rates at different times.

|  |  |  |
| --- | --- | --- |
|  |  | (2.3) |

The linear trend model is expressed by the equation (2.4):

|  |  |  |
| --- | --- | --- |
|  |  | (2.4) |

The above equations are modified using Laplace to allow flexibility in the model and create a generative model. This generative model helps in predicting trends as opposed to the constant trend in conventional models. This study uses the Fourier series for estimating the seasonal part of the model. The study constructs matrices of seasonality vectors for each value of t in historical and future data. The study assumes the effect of holidays to be independent and makes it an additional component of the model. This study uses Stan’s L-BFGS (Byrd et al., 1995) to find a maximum posterior estimate. The study tests the effectiveness of the model for monthly, quarterly, biannual, and annual forecasts. MAPE has been used as a preferred method for this study. While evaluating the forecasts, the driving concept is to make errors understandable for business users to allow them to configure model parameters. The study uses simulated historical forecasts based on “rolling origin” forecast evaluation procedures(Tashman, 2000). It improves the computation times of the above method by using a small sequence of cut-off dates. Prophet (Taylor and Letham, 2017) has lower prediction errors across all periods under observation. The correlation of the estimates of errors is proportional to the number of simulated forecasts. Prophet can provide an easily configurable model with dedicated parameters for trend and seasonality. The analysts can apply domain expertise to determine trend change points and possible growth or movement changes. The holiday parameter can be adjusted based on the problem statement to study the impact of events on time series forecast. The model has not discussed the possibility of anomalies, impacts of unforeseen events on the business. Investor portfolios are generally diversified based on the individual risk appetite. With a variety of asset classes available for investors, it becomes a challenge to allocate between different asset classes and predict the portfolio returns. (Madhuri et al., 2020) uses the prophet model described in a previous study and to enable the user to make predictions by adjusting inflation rate and prevailing bank interest rates. The study uses data from Mar 1981 – Mar 1991 for model training and April 1991 – March 2001 for testing. The data is converted into a python dictionary format. The model is an additive model instead of the usual non-linear models used by most of the above-listed studies. The “pystan” library has been used for building the model. Log transformation is performed on the “y” column to remove non-stationary components. The forecast is built by considering weekly seasonality and holidays. This study presents the outputs of 10 different sectors in graphical format. The study concludes that prophets can identify and predict the trend and the seasonality components. The study has not considered transformation methods other than log transformation for dealing with the stationarity of data. The study has not enlisted or used any concrete statistical evaluation metrics for testing the model. The study has not listed the ten sectors used and hence the study is not generalizable. The impact of events or holidays is not provided as a parameter.

## Studies based on technical indicators

Due to the random walk behaviour of the stock market, the accuracy of traditional regression models is not very good for the prediction of the stock market data. However, it is also known, that availability of additional parameters, feature engineering can help improve the accuracy of machine learning models. (Pahwa and Agarwal, 2019) explore the possibility of improving the performance of linear regression for stock market prediction. To improve the efficiency of the regression model this study proposes to use four features listed below for making the predictions - Adjusted Closing value, the percentage change between a high and low value of the day w.r.t adjusted closing value, the percentage change between adjusted open and adjusted closing price w.r.t to the adjusted opening price, Adjusted volume for the stock. For the processing of the data, the study removes all the null values and scales all the features between –1 and 1. A simple linear regression model is built to make the predictions and the results are evaluated using Adjusted R-squared, RMSE value. This study shows that there is a slight improvement in the prediction accuracy of the model. This study has not considered the autocorrelation, trend, and non-stationary nature of the data. The features used in the study are intercorrelated and the study does not analyze the individual impact of various derived features on the predictive power of the linear regression model.

(Rosita et al., 2021) proposes to use the CNN algorithm for stock market prediction. CNN is known to have zoom-resistant angles and spatial transformations (Rosita et al., 2021). This study proposes to use four technical indicators – RSI, CCI, ATR, MACD as inputs to CNN in addition to a closing price of the stocks. The designed system was able to obtain an accuracy of only 14.01% after training for 100 epochs.

Deep learning helps in designing an appropriate network to approximate these nonlinear relationships. Uncertainty aware attention models have gained popularity in fields of image detection due to their ability to understand the local context and identifying the image patterns. (Gao et al., 2020) proposes to use the uncertainty-aware attention mechanism for time-series prediction. RNN acts as a basic structure for uncertainty-aware attention models. UA contains two RNNs that generate temporal attention and variable attention in terms of time and feature Graphical user interface, application, Teams

Description automatically generatedusing variational interference. The two types of attention weights are assumed as Gaussian distribution with input-dependent noise. When the model is confident in the contribution of the given features, it generates attention with low variance and allocates noisy attention with high variance to unknown features for each input. The structure of the model is shown in Figure 2.3. The data set consists of eight years of data for three different indices representing different kinds of markets in terms of economic growth. CSI300 index from China represents a developing market, New York stock exchange index S&P 500 represents the most advanced financial market, Nikkei225 from Tokyo represents a financial market between developed and developing state. To enhance the performance of the neural networks, this study uses two technical indicators – MACD, ATR, and two macroeconomic variables – Exchange rate and interest rate, in addition, to open close prices of the data. As it is evident from Table 2.2 below, UA-based networks outperform the classical deep learning methods. MLP models are simple neural networks with fully connected layers and hence underperform on all three indices.

Figure 2.3 Uncertainty-aware Attention (UA) model structure (Gao et al., 2020)

Table 2.2 Metric for UA attention models (Gao et al., 2020)

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Dataset** | | **SP500** | | | **Nikkei225** | | | **CSI300** | | |
| **Metrics** | | **RMSE** | **R** | **MAPE** | **RMSE** | **R** | **MAPE** | **RMSE** | **R** | **MAPE** |
|  | MLP | 44.5137 | 0.975 | 0.012 | 250.976 | 0.966 | 0.009 | 61.1055 | 0.98 | 0.013 |
| Models | LSTM | 35.4955 | 0.98 | 0.01 | 228.876 | 0.972 | 0.008 | 49.5815 | 0.988 | 0.01 |
| CNN | 25.7888 | 0.989 | 0.007 | 215.261 | 0.975 | 0.007 | 46.6316 | 0.989 | 0.009 |
|  | UA | **25.4851** | **0.989** | **0.007** | **209.972** | **0.976** | **0.007** | **45.9357** | **0.989** | **0.009** |

Since the UA models are based on multiple RNN models, the study has not considered discussing the enhancements provided by it over individual RNN models. The study has not considered multi-step ahead forecasts for evaluating the performance of UA models.

(Krauss et al., 2017) investigates the performance of deep learning methods against random forests and gradient-boosted trees for S&P 500 from December 1992 until October 2015. The trading strategy involves buying top 10 stocks and selling the bottom 10 stocks based on predicted probabilities, all with equal monetary weight. The above study finds random forests generating higher returns of around 0.43% pre-cost. To extend the above, (Fischer and Krauss, 2018) evaluates the performance of LSTM for the same data and setup and finds it to outperform all the above models.(Ghosh et al., 2020) proposes to evaluate the performance of the above models in a multi-feature setting.This study proposes to conduct the above experiments on data set containing consists of 29 years of Open high low close volume (OHLCV) features starting from January 1990 till December 2018. Data is divided into train: test ratio of 3:1 using a 4-year window and 1-year stride. Three additional features – Intraday returns, returns w.r.t to last closing price, returns with w.r.t to the opening price have been derived. While random forest only uses the above three signals, LSTM is supplied the last n timesteps to leverage the memory features to above to predict returns for (n+1) day. The study uses Robust Scaler Standardization to make data robust to outliers using an interquartile range. At each time t, the stock is divided into two classes based on intraday returns. A cross-sectional median intraday return is calculated for all stocks and is used as a boundary line for deciding the class of stock at a time t. This study uses CuDNNLSTMs (Chetlur et al., 2014) to train LSTM to leverage the power of GPUs. This study employs a strategy is to buy the top ten stocks and short sell the bottom ten stocks. The model tries to predict the probability for each stock to outperform the median intraday return. The study selects twenty stocks based on these probabilities. The experiments conducted in this study conclude that multi-feature settings outperform the single feature settings for the same algorithms. LSTM shows an improvement of 0.23% in daily returns for multiple feature settings as compared to single feature settings. LSTM outperforms the random forest mechanism. The study has restricted its feature extraction to percentage returns w.r.t opening/closing prices. However, advances in statistics have shown that various technical indicators can help in decision-making.

Iranian stock market has been getting increased traction with the growth of the TEDPIX index. The privatization of state-owned firms has led to increasing interest of private and individual investors in the Iranian stock market. Being an evolving stock market, the market has placed certain restrictions on dealing prices. The ±5% limit of the opening price of indices hinders the abnormal market movements in case of unknown events. However, there have not been many studies on the Iranian stock market's stock price movement and forecast using newer machine learning and deep learning approaches.(Nabipour et al., 2020) conducts a detailed analysis of various machine learning and deep learning models and the impact of technical indicators on these models in the context of the Iranian stock market. Stock indices are derived from stocks with high market investments. Since these are a combination of stocks, prediction of the groups is a more challenging task. This study uses data from November 2009 to November 2019 for four stock market groups – Financials, Petroleum, basic metals, Non-metallic minerals. The study uses IQR to detect and eliminate outliers. In addition to time-series data, the study derives ten statistical indicators shown in Figure 2.4 to augment the dataset. The data is then scaled between 0 to 1. The study evaluates two major types of models – tree-based models and neural network models. Performance of the various models is evaluated using key regression metrics – MAPE, MAE, RRMSE, MSE. This study tests the predictions of the models for 1, 2, 5, 10, 15, 20, and 30 days ahead forecasts. The study uses a 70-20-10 split for training, validation, and testing of the data after shuffling. The study concludes that for tree-based models, Adaboost Regressor is a clear winner for Diversified Financials and petroleum sectors. For metallic and non-metallic minerals performance of all 3 boosting techniques is comparable. However, since Adaboost is the fastest regressor, this seems to be an ideal choice for all scenarios. Among neural network models, LSTM outperforms the rest. However, the run time is significantly higher. LSTM outperforms tree models as well. Although this study presents a very detailed analysis of the performance of various models on various stock groups, there are two major gaps in this study - The sampling of data into train, validation, and test data is done on a random basis.Diagram, text

Description automatically generated This is not very accurate in terms of time series as autocorrelation is an important component for time series prediction. This sentiment based on news or events may carry on for few days to weeks. The volume of the stock has been left out of the picture. Volumes of the stock indicate the interest of investors and traders in the stock and impact the direction of the stock in upcoming trading sessions. In easier terms, markets are trying to factor in some news or expectations whenever there are abnormal volumes. Hence this is an important indicator. (Ayala et al., 2021) proposes to build a novel trading strategy by combining deep learning techniques and technical analysis. The study is based on the adaptive market hypothesis (Urquhart and McGroarty, 2016) and considers asymmetric redistribution to enhance the prediction results. The experiments are conducted on three major indices – DJI, IBEX, DAX. (Ayala et al., 2021) propose to use technical indicator TEMA (triple exponential moving average) in addition to commonly used MACD and EMA. This study proposes to create a hybrid strategy using technical indicators and machine learning models. The signals generated by technical indicators are enhanced by the output of the machine learning models. Evaluation of the trading strategy is done in terms of profit-loss ratio, net profit, and average profit per trade. Both components of the strategies are trained separately. The optimized parameters are used to create the hybrid strategy. The novel strategy is again back tested for the final tuning of the parameters. The study observes that strategies based on MACD, TEMA are only profitable on the DJI. However, after adjusting for taxes, these strategies may not be profitable. Also, the risk to reward ratio is higher. Contrary to the usual findings reported in the literature, the experiment results conclude that linear models achieve the best result for given data followed by ANN. The study reports that hybrid models improved profitability on the DJI index and resulted in profitable strategies on the IBEX index. However, the drawdown for the IBEX index is still higher than net profits. The study has not considered the trend information as part of the modelling process. Technical indicators which were chosen above are lagging indicators and prone to false signals due to market volatility.

Figure 2.4 Technical indicators (Nabipour et al., 2020)

As shown by (Granger and Terasvirta, 1993; Franses and van Dijk, 2000; Zhang, 2001), the approximations of time series using linear models are inadequate due to the nonlinear behaviours of the time series. TAR, STAR, ARCH, GARCH are some of the popular statistical models for representing the nonlinear behaviour of the time series. However, these models are not universally applicable as they restrict the type of nonlinearity in the data to empirical characteristics of the data based on the available information (Granger and Terasvirta, 1993). In contrast to plain linear models, hybrid models that combine dissimilar models be more accurate and stable. Neural network mechanisms do not require prior assumptions about the nature of the time series and are more generalizable, leading to improved understanding of the structure, correlation, and hence better forecasts. Time series forecasting with neural networks is treated as an error minimization problem. This leads to many parameters and impacts the interpretability of the models. The selection of the models based only on the reduction of errors is not optimal. The models should be stable, coherent with the time series, and consistent with the previous knowledge (Andrea Sánchez-Sánchez et al., 2020). As stated by (Zhang et al., 1998), there are inconsistent reports of the performance of ANNs. (Zhang et al., 2001) asserts impact of the training set on the performance of the network is limited. The results mainly vary due to forecasting methods, the time horizon of prediction, network structure, training methods, sample data, and model selection parameters. The number of parameters to be selected experimentally is very high. The literature review also reveals the importance of the number of input parameters to the model. (Tang et al., 1991) identifies the advantages of using a large set of input variables. (Zhang et al., 2001) asserts that the number of parameters is more important than the internal structure. As demonstrated by multiple studies referred above recurrence-based neural networks outperform the ANNs. The recurrence function allows forward and backward connections (recurrent or feedback), creating loops within the network architecture by using previous states as a foundation for the current state and preserving an internal memory of the data's actions, facilitating the learning of complex relationships. Their key critique, however, is that they must impose an efficient training algorithm that allows them to capture the dynamics of the sequence, although its use is computationally complex (Andrea Sánchez-Sánchez et al., 2020). The debate about whether pre-processing aimed at series stabilization is important in non-linear models, and even more so in neural networks, is still ongoing, and depends in large part on the type of data being modelled. In the first instance, neural networks' abilities enable them to avoid pre-processing through data transformation. However, it is unclear if, in the case of proper network construction and training, a prior process of removing seasonal trends and patterns is needed. Scaling is often preferable because it reduces training patterns and improves performance.

Literature review reveals that LSTM is capable of modelling non-linear relationships in the data and has considerable accuracy for stock price prediction. Various studies have tried to combine sentiment analysis techniques based on Twitter or news data for improving the modelling capability. A prophet is capable of modelling seasonality in the time series data. (Nagesh, 2021) proposes to build a hybrid model by leveraging prophet and sentiment analysis to improve the accuracy of LSTM models. (Nagesh, 2021) proposes to build an ensemble model by combining Prophet and LSTM model. Predicted prices from the Prophet model are provided as input to the attention-LSTM model in addition to the technical indicators and sentiments derived from news and Twitter data. The Prophet model adjusted for Indian holidays with singular data achieves a validation MAPE value of 11.67. The study has performed a wide range of

Table 2.3 Experiments and Results for the combination of Facebook Prophet and Atten-LSTM (Nagesh, 2021)

|  |  |  |  |
| --- | --- | --- | --- |
|  | Model | Dataset | MAPE |
| Baseline Model | Facebook Prophet | NIFTY 50 Stock Price dataset | 11.67 |
| Proposed Model | Combined  Facebook  Prophet and Atten-  LSTM | Historic Stock Price Data, Tweets Sentiment Polarity, and Facebook Prophet Predictions | 6.72 |
| Historic Stock Price Data, News Article Sentiment  Polarity, and Facebook Prophet Predictions | 5.99 |
| Historic Stock Price Data, Technical Indicators, and Facebook Prophet Predictions | 4.03 |
| Historic Stock Price Data and Facebook Prophet Predictions | 3.68 |
| Historic Stock Price Data, Tweet Sentiment Polarity,  Technical Indicators, and Facebook Prophet Predictions | 5.02 |
| Historic Stock Price Data, News Article Sentiment  Polarity, Technical Indicators and Facebook Prophet Predictions | 4.97 |
| Historic Stock Price Data, News Article Sentiment  Polarity, Tweet Sentiment Polarity, and Facebook Prophet Predictions | 7.63 |
| Historic Stock Price Data, News Article Sentiment  Polarity, Tweet Sentiment Polarity,  Technical Indicators and Facebook Prophet Predictions | 5.19 |

experiments as shown in Table 2.3. The prophet model has outperformed other combinations used in the study. It is also seen that adding sentiment analysis to the prophet as input degraded the performance significantly. The study has not considered the impact of technical indicators on Prophet as a stand-alone model.

Auto-regressive models have been aggressively used for a wide range of time series problems. The efficiency of AR models is driven by strong assumptions about the data. However, for time series with long-range dependencies, the AR models become intractably slow. Autocorrelations also start to vanish. This problem of scalability is solved by CNN and RNN models. As evident from the literature, a neural network-based model is more effective in approximating nonlinear relationships as these do not require any restrictive assumptions. (Triebe et al., 2019) proposes a novel mechanism to bridge the gap between classic-AR and feed-forward neural networks. This study attempts to show that deep learning models can be interpretable, fast, and easy to use for time-series predictions. This study proposes two types of neural network architectures – AR-Net and Sparse AR-Net. AR-Net replicates the AR process using a neural network with no hidden layers. The AR-Net Model uses neural networks to imitate the standard AR process. The AR-coefficients are equivalent to the first layer's parameters. Multilayer model can also be used but it impacts the interpretability. Figure 2.5 and Figure 2.6 show the schematic diagram of an AR-Net Model.

|  |  |
| --- | --- |
| Diagram  Description automatically generated | Diagram  Description automatically generated |
| Figure 2.5 An AR equivalent architecture. (Triebe et al., 2019) | Figure 2.6 An AR inspired neural network with multiple layers. (Triebe et al., 2019) |

Hidden layers are omitted to ensure inter interpretability. To allow the results to be comparable with the classic-AR, MSE is chosen as the evaluation metric. To remove the constraint of knowing the AR-order before training, the model should be fit with sparse AR-coefficients. This study proposes to add a regularization term to achieve the same. It is represented by the equation (2.5) and the regularization is defined as in equation (2.6). The regularization does not discourage large weights like L1-norm or L2-norm. This is important to ensure that actual weights are not regularized to be smaller than their unregularized optima and hence are equivalent to their AR coefficients. Parameters of the above regularization depend on the range of AR coefficients.

|  |  |  |
| --- | --- | --- |
|  |  | (2.5) |

where:

the estimated sparsity of the AR coefficients, user defined.

Regularization strength depends on the estimated noise of data.

|  |  |  |
| --- | --- | --- |
|  |  | (2.6) |

This study aims to mimic the behaviour of the classic-AR model using a neural network. Therefore, the precision of fitted AR-coefficients is used as a key evaluation metric. A One-step-ahead forecast is used as a second evaluation metric. The experiments confirm that the performance of AR-Net and classic-AR had no difference. For Sparse AR-Net the precision of the learned weights was superior to those of classic AR in all scenarios. The AR-Net model was also able to avoid overfitting the noise in the dataset in contrast to the classic-AR. Defying the general understanding that statistical methods outperform neural networks for smaller datasets. The comparison of sparse AR-Net to classic AR is not fully accurate as there are sparse implementations of classic-AR reported in the literature. Although they are computationally expensive, the predictive performance is worth investigating.

## Studies based on sentiment analysis for the stock market.

Numerous external factors impacting the movement on daily basis are a major reason for the random walk nature of the stock market. This leads to uncertainty in prediction. (Nayak et al., 2016) tries to solve this problem by proposing to combine the sentiment analysis from Twitter data with the price movement of the stock. This study uses data set consisting of three companies from different sectors (banking, mining, oil) listed in the Indian stock market. This study builds a daily prediction model by combining a few features derived from historical data with sentiment analysis inputs from news and Twitter data. One of the key derived features is a relationship between the trend and volume of the stock. The trend is derived by considering past n days pattern of up/down. For the monthly model, the trend is calculated based on n months of data. The sentiment analysis is performed in two steps - Lemmatization and assigning polarity to each news/tweet item using a precompiled dictionary with Hadoop Distributed File System (HDFS) and hive. For each item net polarity is the sum of polarities for each word.

Figure 2.7 Prediction Model for Daily Prediction Model (Nayak et al., 2016) Graphical user interface

Description automatically generated with low confidence

For historical price analysis, the authors have considered two prominent patterns – trend and volume. The trend is a pattern of the direction(up/down) of the closing price w.r.t previous closing price for five continuous trading days. The volume of stock traded on daily basis is available from exchanges. For the monthly prediction model, a 24-bit vector is considered as a single data point. The data is trained using these vectors to predict the trend for a complete month. Three machine learning models – boosted decision trees, logistic regression, support vector machines are trained for both daily and monthly prediction models.

The results of the study indicate different models performing differently on different stocks. Table 2.4 shows the summary of results for the daily prediction model. The study provides the prediction vs actual comparison of the monthly data in a diagrammatic format. It is concluded that different models perform better in different sectors. However, there may be days with no news or Twitter buzz for a stock or related sector. The study does not perform any feature reduction and selection before training the models. The neural network models like RNNs may prove more effective for vectorized time series data as prepared in this study. However, the study decides to stick to traditional machine learning models.

Table 2.4 Accuracy of the daily prediction model (Nayak et al., 2016)

|  |  |  |  |
| --- | --- | --- | --- |
| **Model Used** | **Bank** | **Mining** | **Oil** |
| Boosted Decision Tree | 0.458 | 0.78 | 0.769 |
| Logistic Regression | 0.654 | 0.61 | 0.442 |
| Support Vector Machine | 0.51 | 0.59 | 0.442 |

The development of advanced techniques in natural language processing and the availability of news and information online has enabled investors to predict market trends more effectively. However, ease of access has also led to irrelevant, low-quality news or even rumours spread easily. Filtering relevant information for decision-making is an important yet tedious activity. Three key principles derive decision-making for individuals when dealing with such situations – sequential content dependency, diverse influence, and effective and efficient learning (Hu et al., 2017). A novel hybrid attention network (HAN) is a novel algorithm proposed by (Hu et al., 2017) for predicting stock trends based on a sequence of recent related news. To begin, HNN uses attention-based recurrent neural networks (RNN) at a higher level to mimic human cognition in a sequential sense with multiple attentions. The attention system can recognize more influential periods of the sequence, and the RNN layout allows for the processing of recent relevant news for a stock in a single sequence. To model the diverse impact of news, the study proposes to build news-level attention-based neural networks at the lower level, to distinguish more relevant news from others at the same time. The data set consists of three years of Chinese stock market data. The study concludes that the newly proposed HAN model achieves the best accuracy of 0.478 and the highest annualized return of 0.611. The MLP and RF models were at the bottom of the performance metrics. HAN model trained by self-paced learning outperforms HAN with the standard training process.

Previous research mainly focuses on the news articles explicitly specifying the target financial instruments. However, sector-related news can have an equivalent impact on the stock price movement. This study proposes to fix this gap by developing a system to enrich existing prediction models by extracting information from news without explicit mentions for a stock. News distilling network (NDN) (Tang et al., 2021) is a novel algorithm that uses collaborative filtering and neural representation learning to capture the relationship between stocks and news. NDN learns the latent stock and news representations to enable similarity measurements. A gating mechanism is then applied to filter noise. The models are compared using accuracy, Matthews correlation coefficient (MCC) and by creating a buy, hold strategy. The experiments conclude that the NDN model significantly outperforms the hybrid attention network (HAN) model (Hu et al., 2017).

## Studies based on time series similarity

Over a short duration, technical indicators have proven to be suitable. Multiple studies have shown ANN, SVM, genetic programming to be well-suited for financial data modelling. However, stocks from different sectors tend to move together due to similar market conditions. (Sidi, 2020) tries to improve the accuracy of the prediction models by including information about similar stocks. The study is performed on seven stocks from S&P 500 index over five years. Data is split into “n” equal width folds. This study also makes use of commonly used technical indicators – MACD, RSI, volume, open-close spread, etc. This study performs data segmentation using SAX. Further data normalization is performed on a subset of features. The data is modelled using the time window approach for training purpose only. The stock similarity data is not used for testing. Two ensemble models are considered for evaluation – Random Forests and Gradient boosted trees. The study conducts two experiments to study the impact of stock similarity and different kinds of stock similarity functions on stock market predictions. All possible combinations for both experiments will be derived from the set shown in ­­Figure 2.8. Experiment one focuses on feature evaluation, segmentation methods, and temporal modelling window size. It chooses the Euclidean similarity function to pick the top ten similar stocks. The best models from experiment one serve as base models for the second experiment. Experiment two evaluates the impact of five similarity functions – Euclidean distance, Pearson correlation coefficient, Dynamic Time Warping, MINDIST, Co-Integration. It also focuses on various combinations of length fixing functions and the number of similar stocks taken while training. The study by (Sidi, 2020) derives some interesting conclusions.

* Univariate Modelling with SAX transformation is most effective for the prediction of stock market pricing.
* Predicting the rate of change price (PROC) results in better performance than predicting the closing price.
* Classification models have high accuracy but also high standard deviation. This leads to net negative mean returns for the strategy. However, the regressors with lower accuracy have positive mean profit.
* The models trained only on target stock performs better than the model trained on ten similar stocks with Euclidean distance.

However, the study has not considered the amount of data required for training. The study has only considered daily data for five years. The proposed pipeline may perform better if there is a higher volume of data. For this purpose, intraday data can be used instead of daily data. The study has not considered the correlation of the stocks in the S&P which is generally high. This may impact the generalizability of the model. For this purpose, some non-related investment instruments like gold or exchange rates should be used to make the model generalizable and prevent overfitting.

Graphical user interface, diagram, Teams

Description automatically generated

­­Figure 2.8 A configuration tree of all the experiments performed by (Sidi, 2020)

The study has not considered deep learning models like RNN, LSTM that have proven to be more efficient for modelling autocorrelation in time series. Traditional data mining methods have certain limitations in data clustering and prediction. These fail to achieve high accuracy for noisy, random, and nonlinear financial time series. DBSCAN algorithm solves these problems by clustering different nodes based on similarity measurements between the nodes. However, the DBSCAN algorithm is not effective for clustering datasets of different densities since the parameters are sensitive and global. This study aims to improve the DBSCAN algorithm and employ the generalization ability of SVR for making predictions for clusters created by an improved algorithm. (Huang et al., 2019) proposes an enhancement over the DBSCAN algorithm, termed as OVDBSCAN. OVDBSCAN is an unsupervised clustering algorithm obtained through the following proposed enhancements to the DBSCAN algorithm.

* Optimization of the initial point for clustering like the optimizations on the initial point for the k-means algorithm.
* A cyclic clustering process is proposed to further optimize the values of parameters.
* The study uses particle swarm optimization is used to optimize the SVR algorithm.

OVDBSCAN generates n clusters and suppresses the noise and nonstationary components of the data. SVR is then trained on each cluster. Test data is then fed into each trained version of SVR. Experiments prove that the OVDBSCAN algorithm is more effective in the case of uneven data distribution. OVDBSCAN helps in finding many similar classes that improve the regression prediction abilities of SVR. The hybrid OVDBSCAN/SVR model is referred to as the HOS algorithm (Huang et al., 2019). The study also concludes that the new hybrid algorithm has better predictive power for both use cases – the prediction of financial indices and in predicting the trading limits of the stocks for the next day. Although, the algorithm has improved metrics in terms of prediction the running time of the process is much larger. The study has not considered alternate sampling techniques to optimize the run time of the algorithm.

## Summary

This section covers a comprehensive literature review conducted to understand the challenges being faced in stock market prediction and the approach being taken to address those challenges. The Literature review has majorly focused on the following types of studies:

* Studies focusing on plain time series modelling for stock market data using state-of-art models and novel methods.
* Studies using derived technical indicators for improving model performance.
* Studies using sentiment analysis for augmenting the data used in predicting the stock market prediction.
* Studies using stocks with similar time series structure as an additional input to prediction models.

It is also seen that various authors have treated the stock marketed prediction problem as a classification problem instead of regression.

Some major gaps identified during the literature review are listed below:

* Choice of dataset – Numerous studies have used datasets sampled over a single time horizon for making predictions over different time horizons. However, in practice depending upon the type of investment or trading, the different time horizons are used by traders for analysis and have proven to be more effective.
* Use of similar statistical indicators – Technical indicators used in the literature are correlated to each other. The reviewed literature has focused on deriving percentage returns w.r.t different base points. This correlation increases complexity while not offering significant improvements in predictions.
* Most of the novel methods are either extension of neural network models or ensembles of various machine learning models. Although this has improved the accuracy of predictions, the computation times are very high. Also, the models are not interpretable due to an increase in the number of parameters.
* Studies based on classification have shown higher accuracy of predictions, but the strategies built on top of these have been less profitable.
* The models have been tested and developed on developed markets and are not effective in developing markets like India. This raises questions on the generalizability of the models.

This study focuses on addressing the gaps through the following mechanisms:

* The study will remove the dataset bias by choosing a dataset sampled in a daily timeframe and will focus on closing price prediction on daily basis.
* The study will be exploring various technical indicators and rationalize the selection of indicators. It will study the impact of diverse technical indicators on model performance.
* The study aims to address the issue of high complexity for neural network models by proposing the use of certain novel architectures for time series prediction as discussed in section 3.8.
* The study also attempts to address the generalizability of models by choosing a long-time range dataset for a developing market and considering the dataset from the COVID-19 period.

# Research Methodology

## Introduction

This chapter sets the roadmap to achieve the aims and objectives defined for this research. The chapter describes the nuances of data collection, feature engineering, and exploratory data analysis. It discusses the process for the selection of models and evaluation metrics in the context of the research problem. Further, it lays out the detailed process for model building and evaluation.

## Data Collection

The key objective of this research is to build a model using novel mechanisms and evaluate its performance in the context of the Indian stock market. This study also proposes to assess the impact of technical indicators on the model. To achieve the research objectives, a dataset containing the pricing, the volume information of the stock is required. The data for the same is obtained via a free API (Quandl, 2021). Quandl provides the data from the Bombay Stock Exchange (BSE) with daily frequency from the year 1991 till date. This study obtained the data up to March 18, 2021, before the submission of the research proposal. The complete dataset consists of over 4500 stocks and indices from the Bombay stock exchange during the said period. The technical indicators are derived statistically as discussed in

## Data Profiling

The goal of this section is to develop an understanding of the dataset. The data sourced from Quandl has fifteen features as shown in

Table 3.1. The data consists of both stocks that are listed at present or were listed in the past. This study is only interested in the stocks that are currently listed. Initial profiling also reveals that the date range is not the same for all the stocks. This is due to the different listing dates for companies. The pre-processing steps for managing this are discussed in section 3.6. Since we try to generalize the applicability of novel mechanisms for stock market prediction, we select stocks and indices that are allowed for derivatives trading as these are preferred by traders for short term forecasts.

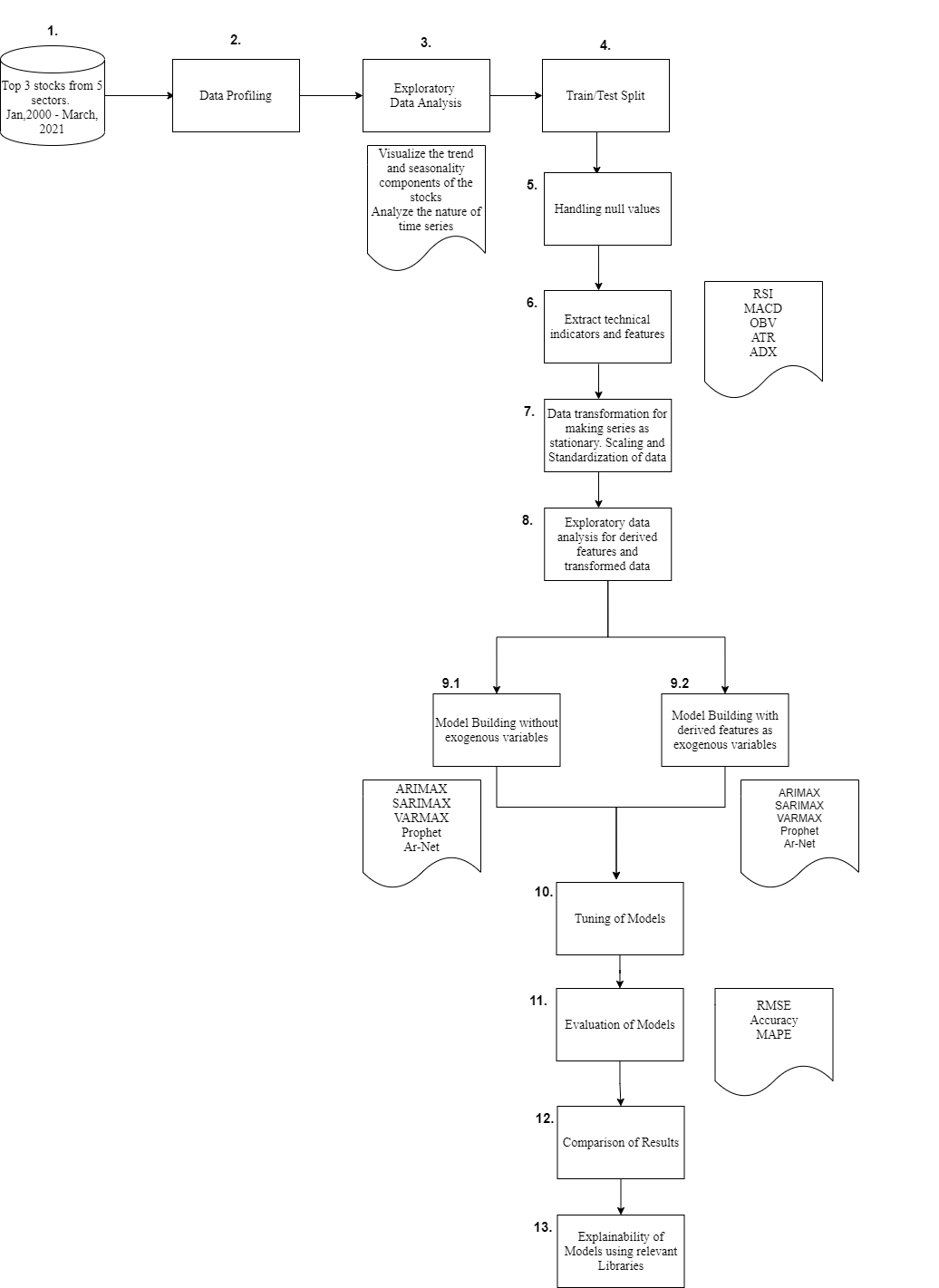


Figure 3.1 Research Methodology

## Initial Exploratory Data Analysis

This section will explore the behaviour of the time series for the selected stocks and indices. The time series graphs will be plotted to visualize the nature and trend of the movement of the stocks. This study will also plot various moving averages over different periods. Based on the

results of (Huang et al., 2019; Sidi, 2020), the similarity between stocks and their respective sectoral indices are explored in this section. The correlation between the daily returns will also be explored between different stocks and indices. Spread H-L for any script is indicative of the intraday volatility of the stock. A visual analysis of the volatility for the different symbols is done to understand the behaviour of the market.

Table 3.1 Features of the dataset.

|  |  |  |
| --- | --- | --- |
| **Id** | **Feature** | **Description** |
|  | Date | Date of the trading day |
|  | Open | First traded price for the stock during the day. |
|  | High | Highest traded price for the stock during the day |
|  | Low | Lowest traded price for the stock during the day. |
|  | Close | The closing price for the day. |
|  | WAP | Weighted Average Price. |
|  | No. of Shares | The absolute number of shares traded for the day. |
|  | No. of Trades | The number of trades for the day for a given script. |
|  | Total Turnover | The total value of the shares traded during the day. |
|  | Deliverable Quantity | The number of shares bought for delivery by traders during that day. |
|  | % Delivery Qty to Traded Qty | Percentage of the traded quantity bought for delivery by investors |
|  | Spread H-L | Spread between the high and the low price for the day |
|  | Spread C-O | Spread between the opening and the closing prices for the day |
|  | Stock Name | Name of the listed company. |
|  | Code | Stock Code for Bombay stock exchange. |

## Train and Test Split

This study exploits the autocorrelation and time-varying behaviour of the data for the modelling process. Therefore, it is important to maintain the order of the records. The data is split into three parts-training, testing, validation set. This study aims to determine the performance of the state-of-the-art models during the COVID19 pandemic. Therefore, this period is as a testing period. The division of the data for the training and validation set is being done based on experiments. The study proposes to begin with an 80/20 split.

## Data Cleaning and Feature Engineering

### Data Cleaning

Quandl syncs its data directly from BSE, the data is inherently clean. However, the following data cleaning steps will be necessary for training the models.

* Dates are formatted to ensure consistency.
* As mentioned, the date ranges of all the stocks are not the same. Since the study aims to generalize the models across sectors, we will drop the dates with null values for any stocks.

### Feature Engineering

The study aims to study the impact of technical indicators on the novel mechanism. This study proposes to use indicators listed in Table 3.2. Indicators have been selected to represent different characteristics of a stock price, namely, volume flow, directional strength, volatility, average price movement. This section will discuss these indicators briefly.

* Relative Strength Indicator (RSI) – The RSI is a technical analysis indicator that examines the intensity of recent price movements to determine if an asset is overbought or overvalued. The value of RSI varies from 0 to 100. A value above seventy is indicative of a possible correction due to being overvalued. A value below thirty indicates an oversold or undervalued state and a possibility of re-rating.
* Moving Average Convergence Divergence (MACD) - The relationship between two moving averages of a security's price is shown by moving average convergence divergence (MACD), a trend-following momentum indicator. A nine-day exponential moving average of MACD is used as a “signal line”. A buy signal is triggered if the MACD line goes above the signal line and vice-versa for the sell signal.
* Average True Range (ATR) – ATR is a measure of market volatility. ATR is directly proportional to the volatility. ATR is based on actual prices; therefore, the range of values will be different for different instruments. This is an important indicator used by traders for exiting the trades.
* Average Directional Index (ADX) – ADX is a measure of the strength of the trend for a stock. It makes use of a positive and a negative directional indicator in addition to the trendline. An ADX value greater than 25 indicates strong strength and a value less than 20 indicates a weak or no trend.

Table 3.2 Technical Indicators

|  |  |  |
| --- | --- | --- |
| **Id** | **Indicator** | **Calculation** |
|  | RSI |  |
|  | MACD |  |
|  | ATR |  |
|  | ADX | Calculation of ADX is a multi-step process referred from (Average Directional Index (ADX) Definition, 2021) |
|  | OBV | If the    else |

* On Balance Volume (OBV) – OBV is a technical indicator based on volume flow to predict changes in stock price. The idea behind this indicator is when volume increases sharply without a significant change in the stock's price, the price will eventually jump upward or fall downward (On-Balance Volume (OBV) Definition, 2021). This indicator generates more actionable insights than simple volume charts. This indicator generates more actionable insights than simple volume charts.
* Percentage returns – This is the percentage change in closing prices for consecutive trading sessions. This helps in models that require stationary data.

### Data pre-processing and transformations

The values of the various stocks are not in the same range. As discussed in sections 3.6.1, 3.6.1, certain derived features are dependent on the actual value of the stock. Based on the results of (Gao et al., 2020), this study proposes to use robust standardization and compare the results with simple log transformation for scaling. The impact of these transformations will be studied without the exogenous variable.

## Detailed Exploratory Data Analysis

Overlapping indicators plots with the price movement is a common practice used by investors and traders for determining trade entry/exits and asset allocation. A common example is a plot of gold to nifty ratio against individual gold and nifty prices. This indicator helps in adjusting the asset allocation among equity and gold and is heavily used by portfolio managers. The dip in the plot indicates a possible rebalance with a higher weight to gold and vice versa. In this section, a similar approach is taken where the technical features derived above are plotted with the respective stock. As discussed in section 3.6.1, the technical indicators reveal different aspects of the respective script. The visual plotting helps in understanding and deriving any possible correlations or patterns. The visual patterns are later verified with the outputs of section 3.11. This helps in understanding the causation of the price movement and the prediction results.

## Model Building

This section discusses the various state-of-art time series prediction models and the novel mechanisms used in this study. Economic or financial variables are not only correlated, with their historical values but also contemporaneously correlated to each other. Various studies cited in Chapter 2 have shown that using correlation of financial variables helps to improve the predictive power of the machine learning models. As a key objective of this study is to study the impact of technical indicators on stock price prediction, this study will focus on machine learning models that can be trained with and without exogenous variables as additional input parameters. This study uses Prophet and AR-Net as novel mechanisms for stock market prediction. The prophet is an additive model, and the auto-regressive neural network is an enhancement over prophet and ARIMA models. Therefore, this study will be using classical machine learning models for benchmarking performance.

The ARIMA model forms the base for the machine learning and novel methods considered in this study. This section discusses the key components of the ARIMA model that apply to the following machine learning models.

### Auto Regression

In an autoregressive (AR) model, the model forecasts the next data point based on previous data points and a mathematical formula similar to that used in linear regression. An order usually represented by the term determines the number of previous data points used for the model. A higher value of means using the data points that occurred a long time ago. This study evaluates the performance of models for long-range dependencies to study the impact of economic cycles on stocks over a long-time horizon.

### Moving Average (MA)

In essence, a MA model states that the subsequent observation is the mean of all previous observations. It helps in accounting for the short-term autocorrelation. The calculations for a moving average model are based on the noise in the data along the data’s slope.

### Trend Differencing(I)

Trend differencing is the number of nonseasonal differences needed for stationarity. This helps in making the data stationary by removing the trend component.

### ARIMAX

The autoregressive model with exogenous variables is a generalization of the ARIMA model which can incorporate external variables and model feedback. The ARMAX model is expressed as:

|  |  |  |
| --- | --- | --- |
|  |  | (3.1) |

Where:

Yt: output series,

Xit: ith input time series or a difference of the ith input series at time t,

m: total number of external covariates,

ki: pure time delay for the effect of the ith input series,

ωi(B): numerator polynomial of the transfer function for the ith input series,

δi(B): denominator polynomial of the transfer function for the ith input series,

Nt: stochastic disturbance in the form of an ARMA model. (Li et al., 2014)

ARMAX models require the time series to be stationary. Therefore, the experiment uses percentage returns as the primary input to the model.

### Seasonal Autoregressive Integrated Moving Average Exogenous model (SARIMAX)

Seasonality in time series means when certain patterns or behaviours occur at known intervals. To incorporate the impact of seasonality in certain businesses and its impact on stock price, this study uses the Seasonal Autoregressive Integrated Moving Average Exogenous i.e., SARIMAX model. SARIMAX applies ARIMA models to the lags that are integral multipliers of seasonality. Additionally, it takes exogenous variables as input. The SARIMAX (p, d, q) (P, D, Q, s) model is represented by equation (3.2).

|  |  |  |
| --- | --- | --- |
|  |  | (3.2) |

The argument ‘s’ represents the periodicity of the seasonal data cycles. This is determined using Grid Search CV. This study uses the best performing s value from the SARIMAX model as the base for modelling the seasonality component of the novel methods that follow.

### Vector Autoregressive Moving-Average Processes with Exogenous Regressors (VARMAX)

Vector autoregression moving average with exogenous regressors is an extension of the VARMA model and hence a generalized form of ARMA. Like the ARMA model, this model is also characterized by the and q parameter and an additional parameter for exogenous variables. The VARMAX model is specified by equation (3.3).

|  |  |  |
| --- | --- | --- |
|  |  | (3.3) |

where the output variables of interest, can be influenced by other input variables . The exogenous variables for a time series can be stochastic or non-stochastic. When exogenous variables are stochastic and their future values are unknown, forecasts of these future values are required to forecast the dependent variables' future values. At times, because exogenous variables are deterministic, their future values can be assumed to be known. The exogenous variables for this study are stochastic. This model forecasts the values of exogenous variables using the VARMA model.

### Prophet

The prophet forecasting model is designed to handle common features of business time series i.e., seasonality, cyclicity, outliers and impact of external variables. It is also designed to be easily interpretable to enable individual, who are not equipped with details of modelling, to be able to tune the model (Taylor and Letham, 2017). The model proposed by the prophet has 3 components – trend ), seasonality, holidays and the error term . The original proposition of the model assumes the error term to be normally distributed.

|  |  |  |
| --- | --- | --- |
|  |  | (3.4) |

Seasonality has been modelled as an additive component like exponential smoothing. The model also allows modelling multiplicative seasonality using log transformation. In contrast to the time series models that account for the temporal structure of the data, this method frames forecasting as a curve-fitting exercise. This approach does not need any missing value imputation or outlier treatment. The fitting mechanism is comparatively faster (Chang et al., 2021). Table 3.3 highlights the key pros and cons of the Prophet model.

Stock markets mechanisms use a mechanism known as a “circuit”. The circuit restricts the movement of the stock beyond a certain point in either direction. For the Indian stock market, the circuit varies for each script. The scripts that are part of the derivatives market also have a circuit for a short duration, while for non-derivative stocks the price cannot move past the circuit for the entire trading session. This is akin to nonlinear but saturating growth.

Prophet proposes to model nonlinear, saturating growth via a novel piece-wise logistic growth model given by equation (3.5). This allows analysts to adjust known change points like quarterly results, election results, central bank policies etc.

|  |  |  |
| --- | --- | --- |
|  |  | (3.5) |

Certain sectors like automobile, construction etc. are cyclical. Short-term investors tend to capitalize on the cyclical nature of these stock. Various trend following strategies also tries to capture the cyclical movements. Therefore, any model needs to factor in the cyclical nature of

|  |  |  |
| --- | --- | --- |
|  |  | (3.6) |

the stocks. Prophet relies on the Fourier series to model the periodic effects. The seasonal component is given by Equation (3.6).

Stock price movements are affected by a lot of non-periodic events. The holidays and events can be adjusted by the analysts. The decomposable nature of the model allows the analyst to gain insights into the forecast, besides just producing a prediction. The smoothing parameter can be used for adjusting the trend flexibility. The strength of the seasonality component is also adjustable.

### AR-Net - Auto-Regressive Neural Network for time-series

Auto-Regressive models use observations from previous time stamps to predict the value at the next timestamp. Simple AR models have faced criticism for their performance when dealing with long-range dependencies. AR models are also affected due to the problem of volatility clustering(Chang and Tsai, 2008) for short term forecasting. However, a study conducted by (Ayala et al., 2021) for three stock market indices shows that linear models can outperform complex models if complemented by the right technical indicators. Neural networks treat time series forecasting as an error minimization problem. This leads to many parameters and hence a lack of interpretability. The AR-Net framework combines the best of traditional AR models and neural networks. This model is proposed to work on time-series with long-range dependencies, needed for monitoring fine granularity data. AR-Net is as interpretable as classic-AR but also scales to long-range dependencies (Triebe et al., 2019). Sparse AR-Net is a regularised form of AR-Net. It takes away the requirement for assumptions about AR coefficients. Normalized data requires only one parameter, the estimated or desired sparsity of the AR-coefficients. The noise standard deviation can be used to estimate the regularisation strength. Due to the novel regularization technique, the optimizer should set small weights to zero and should not change the other weights.

Table 3.3 Pros and Cons of Models.

|  |  |  |
| --- | --- | --- |
| **Model** | **Pros** | **Cons** |
| ARMAX | * Predictive elements can be applied to your time series. No longer confined to time and the expected value. | * Doesn’t fit on a limited amount of data. * Exogenous variables have to be of the same length and time series. * Needs extra pre-processing steps for extra variables. * Needs a stationary time series |
| SARIMAX | * Predictive elements can be applied to your time series. * Helps in modelling seasonality in time series. | * Works well for more. * Exogenous variables have to be of the same length and time series. * Needs extra pre-processing steps for extra variables. |
| VARMAX | * Allows forecasting of multiple parallel time series * External variables can be used to augment the data. * Works for non-stationary time series | * Resource heavy process * Comparatively slow |
| Prophet | * Easy to set up and use. * No missing value treatment requirement. * Automatically detects drastic changes in data and adjusts accordingly. | * Reasons for model performance are not visible to the user. Most of the process is black-boxed. * Limited scope for hyperparameter adjustments. * Not well suited for a small amount of data. |
| AR-Net | * Well-suited for long-range dependencies * Lesser training time compared to conventional models for high order lags. * Interpretable results | * Does not support the modelling of covariate time series. * Needs extra processing for trend and seasonal components. |

The actual weights should be on the same scale as the unregularized optima, as they fully represent the AR coefficients. To accomplish this, a large gradient close to zero is used with a more gradual decrease closer to one. Like this, gradients of regularised weights become nearly imperceptible. This behaviour is achieved by using a root and sigmoid transform of the absolute weight values as shown in equation (3.7).

|  |  |  |
| --- | --- | --- |
|  |  | (3.7) |

The parameters c1, c2 in equation (3.7) depend on the range of AR-coefficients. This study will evaluate the performance of this model for a multi-step ahead forecast. The MSE will be used as an evaluation metric to make sure that results are comparable to the classic-AR model.

## Model Validation

This study considers the stock price prediction as a regression problem. Therefore, it uses standard evaluation metrics for evaluating model performance. The choice of individual metrics is driven by the nature of the time series and by recommendations from the literature as discussed with the respective metric. Table 3.4 highlights the key pros and cons of evaluation metrics considered in this study.

### Mean Absolute Percentage Error (MAPE)

MAPE is a measure of the prediction accuracy of the forecasting methods. It is given by the equation (3.8):

|  |  |  |
| --- | --- | --- |
|  |  | (3.8) |

where: At is the Actual value and Ft is the Forecast value. MAPE is independent of scale as it represents a ratio as a percentage. Based on a review by (de Gooijer and Hyndman, 2006), MAPE is an important evaluation metric for this study due to its interpretability (Taylor and Letham, 2017).

### Root mean squared error (RMSE)

RMSE is a measure of the spread of the error. It is given by the equation (3.9), where represents the actual observation, x̂I represents the predicted value and N is the number of observations. RMSE helps in understanding the spread of the errors. RMSE is used as a metric to understand the performance of models for highly volatile stocks.

|  |  |  |
| --- | --- | --- |
|  |  | (3.9) |

This study has considered data for the year 2020-2021 as its test duration. During this duration, the volatility index was trading at an all-time high. (Triebe et al., 2019) in the original proposition of AR-Net have proposed MSE as an evaluation metric for one step ahead forecasts. This study proposes to use RMSE as it offers more interpretability.

Table 3.4 Pros and Cons of Evaluation Metrics

|  |  |  |
| --- | --- | --- |
| **Metric** | **Pros** | **Cons** |
| RMSE | * Helpful in eliminating or adjusting for large errors, since higher weight is assigned to large errors. * High interpretability. | * Sensitive to outliers |
| Adjusted R-squared | * Useful for eliminating low impact variables ensuring simplicity and explainability of results. | * Based on r-squared calculations. |
| MAPE | * Not impacted by outliers | * May lead to undefined values if the actual value becomes zero |
| Noise | * Efficient for determining the stability of the models. | * The calculation can be complex and have hard interpretability. |

### Adjusted R-squared

It measures how much of the variability of the dependent variables can be explained by the model. Adjusted R2 penalizes the model for excessive use of parameters. The calculation for Adjusted R-squared is a two-step process. First, R-squared is calculated as shown in equation (3.10). Adjusted R-squared is then given by equation (3.11). In equations (3.10), (3.11), p is the number of predictors, N is the total sample size, is the actual observation, ŷi represents the predicted value and y̅ is the sample mean. Adjusted R-squared helps us to determine the impact of increasing technical indicators.

|  |  |  |
| --- | --- | --- |
|  |  | (3.10) |
|  |  | (3.11) |

Adjusted R-squared is preferred because it provides a more exact picture of the correlation by accounting for the number of independent variables added to the model against which the stock or an index is tested.

### Noise

The noise metric calculates how much the model's predictions change daily. A model with high variations is difficult to adopt and much more costly to trade than the more consistent one. The noise metric is used in this study to evaluate the stability of the model. Noise is described as the mean of the absolute difference in the predictions of the model.

## Comparison of the results

This study provides a detailed comparison of the models listed in section 3.8. This study performs a multiple set of comparisons. The first set compares the performance of the models without the use of technical indicators. The second set compares the performance of the models with the use of technical indicators. In the third set, a comparison of the best and the worst-performing models of the above two sets.

## Model Interpretability

Traders and investors will be making their trading decisions based on model outcomes. Their hard-earned money is invested based on the model predictions. To develop stakeholder trust, model outputs must be conveyed to stakeholders. Interpretability of models also helps in identifying the causality of the features. This helps in the further tuning of the models. The machine learning and neural network-based models used in this study are inherently black boxes. This study also uses exogenous variables which makes it more important to have a definitive framework for explaining the model outcomes. Table 3.5 highlights the key pros and cons of commonly used tools. Two commonly used frameworks for interpretability are SHAP(Lundberg and Lee, 2017) and LIME (Local Interpretable Model-agnostic Explanations) (Ribeiro et al., 2016). Both LIME and SHAP are surrogate models. They make minor adjustments to the input and test how the forecast changes. This study uses LIME, an additive feature attribution method, for model interpretability.

|  |  |  |
| --- | --- | --- |
|  |  | (3.12) |

The study uses an extension of the original implementation of LIME by (Metzenthin, 2021) for use with time series. LIME is preferred due to its interpretable data representations. The minimization function for LIME is given by equation (3.12). The first term in the equation is the measure of g's unreliability in approximation f in the Pi-defined locality. In the original study, this is referred to as locality-aware loss.

Table 3.5 Model Interpretability Tools

|  |  |  |
| --- | --- | --- |
| Methods | Pros | Cons |
| Python libraries like Keras and Pytorch | * Easily integrable. * Easy to generate a summary of the model. | * Does not provide an in-depth understanding. * Data transfer between the layers is not visible to the user. |
| SHAP | * Explains the relationships between the variables. | * Needs additional resources. * A steep learning curve to understand the process of integration and interpretation of outcomes. |
| LIME | * Visibility and interpretation are better than SHAP * Provides good coverage of input feature to prediction relationship | * Required even more resources than SHAP. |

The last term is a measure of the g model's complexity. Locality-aware loss is minimised while the second term is kept low enough to be interpretable by humans to maintain both interpretability and local fidelity. LIME achieves local fidelity while optimising for locality-aware loss. LIME is a model agnostic and uses linear explainers to approximate the original model’s decision boundary. Also, as discussed by (Interpreting recurrent neural networks on multivariate time series | by André Ferreira | Towards Data Science, 2021), SHAP is not well adapted to multivariate time series data making LIME a preferred choice for this study.

## Summary

This chapter outlined the decision-making process and final decisions are taken to solve the identified problem. It provides the arguments for the choice of technical indicators. A step-by-step action plan and a list of tasks to be completed are also discussed in detail. This chapter highlights the role of visual representation of technical indicators overlapped with price action and their role in final model analysis. It discussed the relevance and benefits of the following models for stock market prediction – ARMAX, SARIMAX, VARMAX, Prophet, AR-Net. It also discusses four evaluation metrics - MAPE, RMSE, Adjusted R-Squared, Noise, and weighed their benefits for the time-series regression problem. Each metrics helps in the evaluation of different properties of the model like efficiency, stability, volatility etc. This chapter also lays out the plan for a comprehensive comparison of all the experiment results to set a benchmark for stock market analysis. Further, it explains the reason for choosing LIME over SHAP as a tool for explaining the model results.

# Analysis and Design

## Introduction

The section explains the insights into the dataset, various pre-processing steps applied to the data. It also discusses the impact of various pre-processing steps on the model building process. Further, it discusses the nuances of various models, constraints and hyperparameter tuning for all models.

## Dataset Preparation

### Data Filtering

Table

Description automatically generatedAs discussed in sections 3.2 and 3.3, the initial dataset had data for over 4500 stocks with 15 features. For this study, the top three stocks by market cap and their weightage in respective benchmark indices were chosen. The final list of stocks along with the range for which the data is available is listed in Figure.

Figure 4.1 List of stocks used for model training.

It was noted the first date of various scripts is different. This is because these companies went public later than the initial time frame considered for this study. Since the data obtained is since the beginning of the trading history of the stock, this study did not trim the date ranges for any stock. All the stocks were listed on the Bombay stock exchange at the time of data extraction therefore the end date for trading data was the same for each script.

### Preparing a data dictionary

The original data set is a collection of data for all stocks in one excel. However, to train the model on different stocks and preserve the time-series nature of the data, a dictionary of data frames was constructed with stock codes as the keys. Later in the process, each entry of the key was utilised for the building of the models.

### Handling the null values

Only two columns had null values: Deliverable Quantity and % Delivery Qty to Traded Qty. They were removed from the dataset because they were not used in the modelling process or in calculating the technical indicators.

### Derived Features

The study derived features discussed in section 3.6.1. Based on the mathematical derivations, the first few values were found to be null for derived features. This happened because of the lagged nature of the indicators where few past observations are required before computing the values of these indicators.

### Elimination of Extra Features

The research objective of the study is to study the impact of the various technical indicators on the models. The features of the dataset other than close were used to derive the technical indicators and were not required anymore. Therefore, all features except close price and the technical indicators were dropped for cleaning the data set. Figure 4.2- Figure 4.7 show the final data set for six of the fifteen stocks.

### Stationary Tests

Augmented Dickey-Fuller Test (ADF) was conducted to study the stationarity of the closing prices of the stocks. Results for the ADF Test are shown in Table 4.1. Contrary to the general notion, it was observed that closing prices of four stocks exhibited stationary behaviour over the long timer horizon.

|  |  |
| --- | --- |
| Table  Description automatically generated  Figure 4.2 Sun Pharmaceuticals final dataset | Table  Description automatically generated  Figure 4.3 HDFC bank final dataset |
| Table  Description automatically generated  Figure 4.4 Tata Steel final dataset | Table  Description automatically generated  Figure 4.5 Tata motors final dataset |
| Table  Description automatically generated  Figure 4.6 Infosys final dataset | Table  Description automatically generated  Figure 4.7 Bajaj Auto final dataset |

### Transformation of variables

This study performed two kinds of transformations on the data. Since some models like ARIMAX and SARIMAX used in this study are proven to be better performing on stationary data, the closing price of the stock was changed to the percentage change to convert the non-stationary time series to stationary. The results of the ADF Test are shown in Table 4.2. The study finds that one level of differencing is sufficient to convert all non-stationary stock time series into stationary. The derived variables were observed to be on different scales. The study transformed the variables using the standard scaler. <Cite the literature.>

Table 4.1 ADF Test results for all closing prices

|  |  |  |  |
| --- | --- | --- | --- |
| **Stock Name** | **ADF Test Statistic** | **P-Value** | **Result** |
| Sun Pharmaceutical Industries | -5.025754864 | 1.97e-05 | Stationary |
| Divi's Laboratories | -3.571372702 | 0.006332504 | Stationary |
| Dr. Reddy’s Laboratories | -0.081725552 | 0.951228592 | Non-Stationary |
| HDFC Bank | -2.223958871 | 0.197664957 | Non-Stationary |
| ICICI Bank. | -2.658944404 | 0.081416074 | Non-Stationary |
| Kotak Mahindra Bank. | -0.712710304 | 0.843429756 | Non-Stationary |
| Tata Steel | -2.408397069 | 0.139380411 | Non-Stationary |
| JSW Steel | -2.317657173 | 0.166336403 | Non-Stationary |
| Hindalco Industries | -2.300991773 | 0.171648305 | Non-Stationary |
| Bajaj Auto | -1.932661219 | 0.316855625 | Non-Stationary |
| Tata Motors | -2.35160707 | 0.155863141 | Non-Stationary |
| Maruti Suzuki India | -0.638649765 | 0.862007467 | Non-Stationary |
| Infosys | -4.892345637 | 3.61e-05 | Stationary |
| Tata Consultancy Services | -1.535961645 | 0.515652603 | Non-Stationary |
| Wipro | -11.79559857 | 9.60e-22 | Stationary |

Table 4.2 ADF Test Results for Percentage Change

|  |  |  |  |
| --- | --- | --- | --- |
| **Stock Name** | **ADF test statistic** | **p-value** | **Result** |
| Sun Pharmaceutical Industries | -51.48915181 | 0 | Stationary |
| Divi's Laboratories | -13.92312242 | 5.25e-26 | Stationary |
| Dr.Reddy's Laboratories | -71.61844936 | 0 | Stationary |
| Hdfc Bank | -54.58075201 | 0 | Stationary |
| Icici Bank | -32.12679789 | 0 | Stationary |
| Kotak Mahindra Bank | -31.78569665 | 0 | Stationary |
| Tata Steel | -22.03110884 | 0 | Stationary |
| Jsw Steel | -72.64601468 | 0 | Stationary |
| Hindalco Industries | -71.52179026 | 0 | Stationary |
| Bajaj Auto | -53.54510407 | 0 | Stationary |
| Tata Motors | -15.02951777 | 9.95e-28 | Stationary |
| Maruti Suzuki India | -64.58763972 | 0 | Stationary |
| Infosys | -32.05424134 | 0 | Stationary |
| Tata Consultancy Services | -14.15125564 | 2.16e-26 | Stationary |
| Wipro | -13.78478693 | 9.15e-26 | Stationary |

### Exploratory Data Analysis

### Univariate Analysis

Line charts along with moving averages and RSI were plotted for all fifteen stocks as shown in Figure 4.9 - Figure 4.22. Some key common observations are listed below.

* All the plots appear to be a random walk with no visible pattern.
* Some of the stocks have shown a sharp fall once or twice during the entire period. Further analysis revealed that this is due to the stock split corporate action for the stock. This confirms that the dataset is not adjusted for corporate actions or fundamental changes to the underlying stock. JSW Steel did a stock split on Jan 4, 2017. Similarly, Hindalco had a stock split in August 2005 and so on.
* Tata Steel shows a cyclical pattern from the year 2010-2020.
* All stocks had a steep decline in March 2021 during the commencement of lockdowns due to COVID-19.

### Multivariate Analysis

To develop an understanding of the importance of technical indicators, a visual analysis of individual stocks was conducted. The time frame that had large up or downtrends were focused to understand the signals given by various technical indicators. In this section, we highlight few such instances.

Figure 4.8 shows the chart for Hindalco Industries Sampled Monthly(HINDALCO, 2021). It was observed that at all four instances – June 1, 2009, Oct 1, 2013, July 1, 2016, Oct 1, 2020, where the MACD line (blue) crossed above the signal line (orange), the stock had a long rally that lasted over months. It was also noted that RSI for the stock at these points lied at the lower end of the range. Similarly on dates – Jun 2, 2008, Jun 1, 2011, Feb 2, 2015, April 2, 2018, the MACD line crosses below the signal line, the stock corrected and almost giving up all the gains since the last rally. RSI at these date points was above or near the upper threshold.

Graphical user interface, chart, line chart

Description automatically generated

Figure 4.8 Hindalco Industries resampled monthly

The combination of the above two movements also indicated the existence of cyclicity in the stock movement. Figure 4.9 daily plot and monthly plot also indicated that the stock was bound in range and did not give significant returns to investors for a long time.

Timeline

Description automatically generated

Figure 4.9 Hindalco Industries daily plot

Figure 4.10, Figure 4.11 show the HDFC bank daily and monthly chart respectively with technical indicators. The steep change on the daily chart is due to the stock split on July 14,

Timeline

Description automatically generated

Figure 4.10 HDFC Bank

2011 (HDFCBANK - Trading View, 2021). Chart, line chart

Description automatically generated As evident in Figure 4.11, HDFC bank had muted price growth for almost ten years. RSI indicator remained in the overbought category for most of the above period, proving the significance of the indicator. The MACD and signal line also appeared to be in a zig-zag pattern with no clear long term trend. Two major trends were observed on the chart – an uptrend from Feb 2017 to Sep 2018. The trend was indicated by MACD crossing above the signal line combined with the strong ADX signal, a downtrend from January 2020 to Oct 2020. The downtrend was pre-indicated by MACD and ADX indicators before the onset of COVID lockdowns.

Figure 4.11 HDFC Bank monthly with technical indicators

Figure 4.12 shows the monthly plot of TCS. TCS depicted behaviour similar to HDFC Bank, being stuck in a range for a long period. The range is also depicted by the ATR indicator. The first major change was in October 2017 which led to a multi-year rally in the stock. MACD signal gave the signal in Septemeber 2017.

Chart, line chart

Description automatically generated

Figure 4.12 Tata Consultancy Services monthly data with indicators

Figure 4.13 shows the daily chart. The daily chart provided better insights into small trends that occurred between the years 2010 and 2015. TCS had three stock splits – 28 July 2006,

Timeline

Description automatically generated

Figure 4.13 Tata Consultancy Services

Jun 16, 2009, and May 30, 2018, during the period in consideration (TCS Trading View, 2021)

|  |
| --- |
|  |
| Figure 4.14 Tata Steel |
|  |
| Figure 4.15 ICICI Bank |
| Figure 4.16 Kotak Mahindra Bank |
| Figure 4.17 Infosys Ltd. |
|  |
| Figure 4.18 Wipro Ltd. |
| Figure 4.19 Divi's Laboratories |
| Figure 4.20 Dr Reddy's Laboratories |
| Figure 4.21 Sun Pharmaceutical |
| Figure 4.22 Bajaj Auto |
| Figure 4.23 Maruti Suzuki India |
| Figure 4.24 Tata Motors |

### Modeling

To study the impact of techincal indicators on the prediction, two experiments were broadly conducted for each proposed model – Training the model only on the closing prices, introducing the exogenous variables to the modelling process. The prior is considered as the base model for each case. Each of the two proposed scenarios were extended to using percentage change as a stationary time series alternative of closing prices. This substitution was enabled by the feasibility of conversion of percentage change into final closing price prediction.

## Summary

# Results and Discussions

## Introduction

## Interpretation of Visualizations

## Evaluation of various methods etc...

## Summary

# Conclusion and Recommendations

## Introduction

## Discussion and Conclusion

## Contribution to Knowledge

## Future Recommendations

# References

Aali-Bujari, A., Venegas-Martínez, F. and Pérez-Lechuga, G., (2014) Impact of the Stock Market Capitalization and the Banking Spread in Growth and Development in Latin American: A Panel Data Estimation with System GMM. *MPRA Paper*. [online] Available at: https://ideas.repec.org/p/pra/mprapa/56588.html [Accessed 8 Apr. 2021].

Abraham, A., Nath, B. and Mahanti, P.K., (2001) Hybrid intelligent systems for stock market analysis. In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. [online] Springer Verlag, pp.337–345. Available at: https://link.springer.com/chapter/10.1007/3-540-45718-6\_38 [Accessed 24 Mar. 2021].

Akhter, S. and Mısır, M., (2005) Capital Markets Efficiency: Evidence from the Emerging Capital Market with Particular Reference to Dhaka Stock Exchange. *undefined*.

Andrea Sánchez-Sánchez, P., Rafael García-González, J. and Haidy Perez Coronell, L., (2020) Encountered Problems of Time Series with Neural Networks: Models and Architectures. In: *Recent Trends in Artificial Neural Networks - from Training to Prediction*. [online] IntechOpen. Available at: www.intechopen.com [Accessed 20 Apr. 2021].

Anon (2021) *Average Directional Index (ADX) Definition*. [online] Available at: https://www.investopedia.com/terms/a/adx.asp [Accessed 11 Apr. 2021].

Anon (2021) *Financial Terms Dictionary*. [online] Available at: https://www.investopedia.com/financial-term-dictionary-4769738 [Accessed 11 Apr. 2021].

Anon (2021) *HDFCBANK - Trading View*. [online] Available at: https://in.tradingview.com/chart/TheG0QSs/?symbol=BSE%3AHDFCBANK [Accessed 23 Jul. 2021].

Anon (2021) *HINDALCO*. [online] Available at: https://in.tradingview.com/chart/TheG0QSs/?symbol=BSE%3ARELIANCE [Accessed 22 Jul. 2021].

Anon (2021) *Interpreting recurrent neural networks on multivariate time series | by André Ferreira | Towards Data Science*. [online] Available at: https://towardsdatascience.com/interpreting-recurrent-neural-networks-on-multivariate-time-series-ebec0edb8f5a [Accessed 11 Jun. 2021].

Anon (2021) *On-Balance Volume (OBV) Definition*. [online] Available at: https://www.investopedia.com/terms/o/onbalancevolume.asp [Accessed 4 Jun. 2021].

Anon (2021) *TCS Trading View*. [online] Available at: https://in.tradingview.com/chart/TheG0QSs/?symbol=BSE%3ATCS [Accessed 23 Jul. 2021].

Ayala, J., García-Torres, M., Noguera, J.L.V., Gómez-Vela, F. and Divina, F., (2021) Technical analysis strategy optimization using a machine learning approach in stock market indices. *Knowledge-Based Systems*, [online] 225, p.107119. Available at: https://linkinghub.elsevier.com/retrieve/pii/S0950705121003828 [Accessed 22 May 2021].

Bustos, O. and Pomares-Quimbaya, A., (2020) *Stock market movement forecast: A Systematic review*. *Expert Systems with Applications*, .

Byrd, R.H., Lu, P., Nocedal, J. and Zhu, C., (1995) A Limited Memory Algorithm for Bound Constrained Optimization. *SIAM Journal on Scientific Computing*, 165, pp.1190–1208.

Chang, B.R. and Tsai, H.F., (2008) Forecast approach using neural network adaptation to support vector regression grey model and generalized auto-regressive conditional heteroscedasticity. *Expert Systems with Applications*, 342, pp.925–934.

Chang, W., Cheng, J., Allaire, J., Sievert, C. and Schloerke, B., (2021) Web Application Framework for R [R package shiny version 1.6.0]. [online] Available at: https://CRAN.R-project.org/package=shiny [Accessed 6 Jun. 2021].

Chetlur, S., Woolley, C., Vandermersch, P., Cohen, J., Tran, J., Catanzaro, B. and Shelhamer, E., (2014) cuDNN: Efficient Primitives for Deep Learning. [online] Available at: http://arxiv.org/abs/1410.0759 [Accessed 16 May 2021].

Cont, R., (2001) Empirical properties of asset returns: Stylized facts and statistical issues. *Quantitative Finance*, [online] 12, pp.223–236. Available at: https://www.tandfonline.com/doi/abs/10.1080/713665670 [Accessed 16 May 2021].

Dauphin, Y.N., Fan, A., Auli, M. and Grangier, D., (2016) Language Modeling with Gated Convolutional Networks. *34th International Conference on Machine Learning, ICML 2017*, [online] 2, pp.1551–1559. Available at: http://arxiv.org/abs/1612.08083 [Accessed 19 May 2021].

Fama, E.F., (1995) Random Walks in Stock Market Prices. *Financial Analysts Journal*, [online] 511, pp.75–80. Available at: https://www.tandfonline.com/doi/abs/10.2469/faj.v51.n1.1861 [Accessed 5 Apr. 2021].

Fischer, T. and Krauss, C., (2018) Deep learning with long short-term memory networks for financial market predictions. *European Journal of Operational Research*, 2702, pp.654–669.

Franses, P.H. and van Dijk, D., (2000) *Nonlinear Time Series Models in Empirical Finance*. [online] *Nonlinear Time Series Models in Empirical Finance*. Cambridge University Press. Available at: https://www.cambridge.org/core/books/nonlinear-time-series-models-in-empirical-finance/FF6720F6B34548290D813D9652FB425A [Accessed 14 May 2021].

Gao, P., Zhang, R. and Yang, X., (2020) The Application of Stock Index Price Prediction with Neural Network. *Mathematical and Computational Applications*, [online] 253, p.53. Available at: https://www.mdpi.com/2297-8747/25/3/53 [Accessed 9 Apr. 2021].

Gerlein, E.A., McGinnity, M., Belatreche, A. and Coleman, S., (2016) Evaluating machine learning classification for financial trading: An empirical approach. *Expert Systems with Applications*, 54, pp.193–207.

Ghosh, P., Neufeld, A. and Sahoo, J.K., (2020) *Forecasting directional movements of stock prices for intraday trading using LSTM and random forests*.

de Gooijer, J.G. and Hyndman, R.J., (2006) 25 years of time series forecasting. *International Journal of Forecasting*, 223, pp.443–473.

Granger, C.W.J. and Terasvirta, T., (1993) Modelling Non-Linear Economic Relationships. *OUP Catalogue*. [online] Available at: https://ideas.repec.org/b/oxp/obooks/9780198773207.html [Accessed 14 May 2021].

Harvey, A.C. and Peters, S., (1990) Estimation procedures for structural time series models. *Journal of Forecasting*, [online] 92, pp.89–108. Available at: https://onlinelibrary.wiley.com/doi/full/10.1002/for.3980090203 [Accessed 10 May 2021].

Hu, Z., Liu, W., Bian, J., Liu, X. and Liu, T.-Y., (2017) Listening to Chaotic Whispers: A Deep Learning Framework for News-oriented Stock Trend Prediction. *arXiv*. [online] Available at: http://arxiv.org/abs/1712.02136 [Accessed 24 May 2021].

Huang, M., Bao, Q., Zhang, Y. and Feng, W., (2019) A Hybrid Algorithm for Forecasting Financial Time Series Data Based on DBSCAN and SVR. *Information*, [online] 103, p.103. Available at: https://www.mdpi.com/2078-2489/10/3/103 [Accessed 9 Apr. 2021].

Ji, S., Kim, J. and Im, H., (2019) A comparative study of bitcoin price prediction using deep learning. *Mathematics*, [online] 710, p.898. Available at: https://www.blockchain.com/charts. [Accessed 19 May 2021].

Krauss, C., Do, X.A. and Huck, N., (2017) Deep neural networks, gradient-boosted trees, random forests: Statistical arbitrage on the S&P 500. *European Journal of Operational Research*, 2592, pp.689–702.

LALOUX, L., CIZEAU, P., POTTERS, M. and BOUCHAUD, J.-P., (2000) RANDOM MATRIX THEORY AND FINANCIAL CORRELATIONS. *International Journal of Theoretical and Applied Finance*, 0303, pp.391–397.

Lam, C., (n.d.) *Challenges to Time Series Analysis in the Computer Age*.

Li, Y., Su, Y. and Shu, L., (2014) An ARMAX model for forecasting the power output of a grid connected photovoltaic system. *Renewable Energy*, 66, pp.78–89.

Lo, A.W., (2004) The adaptive markets hypothesis. *Journal of Portfolio Management*, [online] 30SUPPL., pp.15–29. Available at: https://jpm.pm-research.com/content/30/5/15 [Accessed 12 Jun. 2021].

Lundberg, S. and Lee, S.-I., (2017) A Unified Approach to Interpreting Model Predictions. *Advances in Neural Information Processing Systems*, [online] 2017-December, pp.4766–4775. Available at: http://arxiv.org/abs/1705.07874 [Accessed 9 Jun. 2021].

Madhuri, C.R., Chinta, M. and Kumar, V.V.N.V.P., (2020) Stock market prediction for time-series forecasting using prophet upon ARIMA. In: *2020 7th International Conference on Smart Structures and Systems, ICSSS 2020*. Institute of Electrical and Electronics Engineers Inc.

Metzenthin, E., (2021) *emanuel-metzenthin/Lime-For-Time: Application of the LIME algorithm by Marco Tulio Ribeiro, Sameer Singh, Carlos Guestrin to the domain of time series classification*. [online] Available at: https://github.com/emanuel-metzenthin/Lime-For-Time [Accessed 9 Jun. 2021].

Møller, M.F., (1993) A scaled conjugate gradient algorithm for fast supervised learning. *Neural Networks*, 64, pp.525–533.

Mudassir, M., Bennbaia, S., Unal, D. and Hammoudeh, M., (2020) Time-series forecasting of Bitcoin prices using high-dimensional features: a machine learning approach. *Neural Computing and Applications*, [online] pp.1–15. Available at: https://doi.org/10.1007/s00521-020-05129-6 [Accessed 21 May 2021].

Nabipour, M., Nayyeri, P., Jabani, H., Mosavi, A., Salwana, E. and S., S., (2020) Deep Learning for Stock Market Prediction. *Entropy*, [online] 228, p.840. Available at: https://www.mdpi.com/1099-4300/22/8/840 [Accessed 30 Mar. 2021].

Nagesh, P., (2021) Combination of Facebook Prophet and Attention-Based LSTM Combination of Facebook Prophet and Attention-Based LSTM with Multi-Source data for Indian Stock Market Prediction with Multi-Source data for Indian Stock Market Prediction. *Dissertations*. [online] Available at: https://arrow.tudublin.ie/scschcomdis [Accessed 27 May 2021].

Nayak, A., Pai, M.M.M. and Pai, R.M., (2016) Prediction Models for Indian Stock Market. In: *Procedia Computer Science*. Elsevier B.V., pp.441–449.

Pahwa, K. and Agarwal, N., (2019) Stock Market Analysis using Supervised Machine Learning. In: *Proceedings of the International Conference on Machine Learning, Big Data, Cloud and Parallel Computing: Trends, Prespectives and Prospects, COMITCon 2019*. Institute of Electrical and Electronics Engineers Inc., pp.197–200.

Patel, J., Shah, S., Thakkar, P. and Kotecha, K., (2015) Predicting stock and stock price index movement using Trend Deterministic Data Preparation and machine learning techniques. *Expert Systems with Applications*, 421, pp.259–268.

Quandl, (2021) *BSE | Bombay Stock Exchange | Quandl*. [online] Available at: https://www.quandl.com/data/BSE-Bombay-Stock-Exchange [Accessed 3 Jun. 2021].

Ribeiro, M.T., Singh, S. and Guestrin, C., (2016) “Why should i trust you?” Explaining the predictions of any classifier. In: *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. [online] Association for Computing Machinery, pp.1135–1144. Available at: http://dx.doi.org/10.1145/2939672.2939778 [Accessed 9 Jun. 2021].

Rosita, A., Ahmad, A., Saudi, M.H., Iskandar, S., Suri, S. and Saudi, M.M., (2021) *Market Price Signal Prediction Based On Deep Learning Algorithm*. [online] *Turkish Journal of Computer and Mathematics Education*, Available at: https://www.turcomat.org/index.php/turkbilmat/article/view/5995 [Accessed 23 May 2021].

Sidi, L., (2020) Improving S&P stock prediction with time series stock similarity. *arXiv*. [online] Available at: http://arxiv.org/abs/2002.05784 [Accessed 9 Apr. 2021].

Spearman, C., (1987) The proof and measurement of association between two things. By C. Spearman, 1904. *The American journal of psychology*, 1003–4, pp.441–471.

Tang, T.-H., Chen, C.-C., Huang, H.-H. and Chen, H.-H., (2021) Retrieving Implicit Information for Stock Movement Prediction; Retrieving Implicit Information for Stock Movement Prediction. *Development in Information Retrieval*, [online] p.5. Available at: https://doi.org/10.1145/3404835.3462999 [Accessed 23 May 2021].

Tang, Z., de Almeida, C. and Fishwick, P.A., (1991) Time series forecasting using neural networks vs. Box-Jenkins methodology. *Simulation*, [online] 575, pp.303–310. Available at: https://journals.sagepub.com/doi/abs/10.1177/003754979105700508 [Accessed 14 May 2021].

Tashman, L.J., (2000) Out-of-sample tests of forecasting accuracy: An analysis and review. *International Journal of Forecasting*, 164, pp.437–450.

Taylor, S.J. and Letham, B., (2017) Forecasting at Scale. [online] Available at: https://doi.org/10.7287/peerj.preprints.3190v2 [Accessed 27 Mar. 2021].

Triebe, O., Laptev, N. and Rajagopal, R., (2019) AR-Net: A simple Auto-Regressive Neural Network for time-series. *arXiv*. [online] Available at: http://arxiv.org/abs/1911.12436 [Accessed 14 Apr. 2021].

Urquhart, A. and McGroarty, F., (2016) Are stock markets really efficient? Evidence of the adaptive market hypothesis. *International Review of Financial Analysis*, 47, pp.39–49.

Zhang, G., Eddy Patuwo, B. and Y. Hu, M., (1998) Forecasting with artificial neural networks: The state of the art. *International Journal of Forecasting*, 141, pp.35–62.

Zhang, G.P., (2001) An investigation of neural networks for linear time-series forecasting. *Computers and Operations Research*, 2812, pp.1183–1202.

Zhang, G.P., Patuwo, B.E. and Hu, M.Y., (2001) A simulation study of artificial neural networks for nonlinear time-series forecasting. *Computers and Operations Research*, 284, pp.381–396.

# Chart, bar chart Description automatically generated with medium confidenceResearch Plan

# Research Proposal

# Abstract

With the ever-decreasing bank interest rates, increasing investor awareness and risk appetite, stock market is becoming a preferred investment tool for a common man. However, in addition to the investments, short term stock trading is an important part of financial markets. Majority of the traders use leverage to make short term trades. Due to leverage, accuracy of the trading decisions is key to a success. Over the time, traders have developed various technical indicators to aid the financial decisions. Stock market data is essentially a time series data that exhibits characteristics random walk. Researchers have attempted to maximize stock market prediction accuracy using machine learning and deep learning algorithms as computational resources have become more widely available. However, like in many business problems, forecasting at scale has been a challenge for machine learning algorithms. Interpretability of the deep learning methods has been another key challenge area for the researchers in case of stock market prediction. This study focuses on evaluating novel time series forecasting methods against the state-of-art time series prediction algorithms. The study studies the effect of popular technical indicators on these models. The study also enhances the existing literature by assessing the performance of the state-of-the art machine learning algorithms on the Indian stock market for the years 2020-2021. This research will provide analysts and traders with new methods for making predictions on Indian stock market based on long term dependencies and reduce bottlenecks involved in forecasting at scale.

[Table 2.1 Results summary (Ji et al., 2019) 17](#_Toc75049456)

[Table 2.2 Metric for UA attention models (Gao et al., 2020) 22](#_Toc75049457)

[Table 2.3 Experiments and Results for the combination of Facebook Prophet and Atten-LSTM (Nagesh, 2021) 27](#_Toc75049458)

[Table 2.4 Accuracy of the daily prediction model (Nayak et al., 2016) 31](#_Toc75049459)

[Table 3.1 Features of the dataset. 38](#_Toc75049460)

[Table 3.2 Technical Indicators 40](#_Toc75049461)

[Table 3.3 Pros and Cons of Models. 46](#_Toc75049462)

[Table 3.4 Pros and Cons of Evaluation Metrics 48](#_Toc75049463)

[Table 3.5 Model Interpretability Tools 50](#_Toc75049464)

[Table 8.1:Features of the dataset 81](#_Toc75049465)

[Table 8.2: Technical Indicators 82](#_Toc75049466)

# Introduction

The stock market is an important component of a nation’s economy. Research has shown that stock market capitalization positively impacts the economic growth of a country(Aali-Bujari et al., 2014). Any major political, economic, even natural developments that determine the economic growth of the country also directly impact the stock market movement and vice versa.

Over the last 3 decades SENSEX has delivered approximately 11.74% of CAGR returns. Stock market has been one of the most rewarding yet among the most feared investment avenue.

However, recent break of COVID-19 pandemic that led to nationwide lockdown led to a crash in the market wiping out crores of rupees worth of investor wealth. On the other hand, the sharp recovery of the markets provided ample opportunity for the investors to deploy any idle cash for superior returns.

Other segment of people who benefit from the rewarding nature of stock market are the traders. Traders usually buy/sell leveraged positions in the market for very short duration, ranging from few seconds to a few days. There are roughly 4500 stocks listed on Bombay stock exchange. Since most of the traders take leveraged positions, they prefer to do so in Futures or in liquid scripts only to facilitate higher margins and easy square-off.

Traders use various technical indicators for stock selection and trading. Some of the commonly used technical indicators are – EMA, RSI, MACD, OBV, ATR, ADX. A combination of these is generally used by traders for making trade decisions. Accuracy of the decisions is a key to success for trading system.

# Background and related research

(Fama, 1995) proposed the efficient market hypothesis, which suggested that stock prices behave like a random walk, making it difficult to predict. However, over the time, various attempts have been made to model and predict the stock price movements. Some research is based purely on technical analysis while many researchers tried to incorporate financial or fundamentals of the company into making the prediction. (Nayak et al., 2016) proposes to use statistically derived feature and combine it with sentiment derived from Twitter and news data for Indian stock markets. This study tries to convert the daily data into a 24-bit vector representing a monthly data point and make prediction for complete vector as a classification task. This study concludes that among Boosted Decision trees, Logistic Regression and SVM, Logistic regression performs best for financial sector while Boosted Decision Tree gave better accuracy for Mining and Oil sectors. (Pahwa and Agarwal, 2019) study the movement of Google Stock data for 14 years of data (2005-2018). This study uses only 4 feature – Adjusted Closing and adjusted volume and HL\_PCT, PCT\_CHANGE as the input parameters to linear regression model. The study concludes that use of derived features improves accuracy for simple machine learning models. (Sidi, 2020) enhances the predictability of Gradient boosting based classifier and regressor by exploiting the correlation between movement of certain stocks. This study evaluates 5 time series similarity functions (co-integration, DTW, Euclidean, Pearson, and SAX) for determining relevant predictors. For data processing, it uses different segmentation methods and temporal modeling techniques over 1, 3, 7 days period. Mean accuracy for the enhanced model increased by 0.03 points. (Huang et al., 2019) proposes a novel time series clustering algorithm-OVDBCSAN for clustering of financial time series data. SVR is then trained on clusters obtained by above algorithm. Hybrid algorithm is called HOS algorithm and study finds significant improvements in prediction for data from Shanghai Stock Exchange. (Gao et al., 2020) evaluate the performance of MLP, LSTM, CNN, and UA models on 3 stock market indices of different economies. SP500, CSI300, Nikkei225 represent developed, less developed and developing financial market, respectively. Study also uses 7 technical and external indicators including exchange rate and central bank interest rates. This study concludes that UA based neural networks had best performance in all 3 kinds of markets. (Nabipour et al., 2020) employed decision trees, random forest, Adaboost, XGBoost, LSTM, RNN for prediction of future values of stocks in 4 different groups from Tehran stock exchange. Study considers 10 technical indicators as exogenous variables. Study concludes that AdaBoost regressor is best among considered machine learning methods. LSTM performs best among the neural network models. (Taylor and Letham, 2017) introduce a three-component decomposable time series model: pattern, seasonality, and holidays. Effectively, a time series forecasting problem is converted to a curve-fitting exercise. Interpretability of the model makes tuning parameters easy even for someone who lacks expertise in forecasting models. (Madhuri et al., 2020) use Prophet (Taylor and Letham, 2017) to project portfolio returns based on user’s risk profile and investment instruments. The study uses data from Mar,1981-Mar,1991 as training data and April,1991- March,2001 for analysis. It concludes that by providing different type seasonality components, a generalizable model can be built for complete portfolio analysis. (Triebe et al., 2019) suggest using feed forward neural networks to model AR-process dynamics. This helps in effectively modelling the time series for long range dependencies. (Bustos and Pomares-Quimbaya, 2020) provide a systematic review of state-of-the-art stock market forecast from 2014-2018 for different stock exchanges across the globe. All these studies have treated prediction as a classification problem, trying to predict the direction of movement instead of the price. Among these only (Patel et al., 2015) analysis data for stocks and Indian stock market indices. This study summarizes the performance of different models over the years. (Lam, n.d.) explains various examples in which discovering the patterns and forecasting for long time periods renders traditional methods ineffective. Similarly, (Andrea Sánchez-Sánchez et al., 2020) explains various challenges faced by neural networks in time series prediction especially in cases with high dimensional data and large number of patterns. As the network grows and becomes complex, stability of neural network goes for a toss with even slight changes in data, parameters, configuration etc.

As it is evident from the literature that novel mechanisms have not been tested on the recent data of Indian stock market. Also, no comparison has been done between novel time series prediction algorithms and state-of-art machine learning methods. This study is novel is because it fills above two identified gaps. Also, this research asses the performance of state-of-art algorithms in context of Indian stock market especially for the years 2020, 2021. The movement of the market in these years has been unprecedented due to COVID-19 pandemic and no existing study that provides a benchmark for performance of models during this period was found.

# Problem statement

Research based on technical theories has a basic assumption that history repeats itself (Nayak et al., 2016). Various studies have tried to break the stock market data into seasonality and trend and noise components to study the behavior of time series and make prediction. Studies have shown that for simple machine learning models for time series prediction, the autocorrelation decays exponentially as the time gap increases. Neural networks have been proven to be efficient for stock market data, but the high number of factors and configurations make them unstable technique. Also, these techniques are not explainable. (Andrea Sánchez-Sánchez et al., 2020). AR-Net is a novel method that scales for long range dependencies while tending to keep the model information explainable and computationally feasible. Prophet is known to be a practical approach to forecasting at scale. However, Literature review reveals that no study for benchmarking the performance of traditional time series forecasting models and the novel time-series prediction models - Prophet (Taylor and Letham, 2017), and Auto-Regressive Neural Network (Triebe et al., 2019) for Indian stock market has been done. Since times 2020 and 2021 are unprecedented, this study will also evaluate the performance of the state-of-art time series models for the stock market data including the above-mentioned period and compare the performance of these models with Prophet and Auto-Regressive Neural Network.

# Research Questions

Literature review and the identified gaps lead to following research questions:

* Can the derived technical variables enhance the performance of classical time series algorithms?
* Are the novel algorithms comparable to the state-of-the art time series prediction algorithms?
* Is the novel neural-network framework interpretable yet effective for the stock market prediction?

# Aim and Objectives.

The main aim of this research is to propose a prediction model for stock market prediction. The primary goal of this research is to explore and demonstrate the performance of various state-of-art and novel time series prediction algorithms in context to the stock market data.

The research objectives are formulated based on the aim of this study which are as follows:

* To assess the performance of state-of-art machine learning models for time series prediction against a novel machine learning and a novel neural network method.
* To analyze the performance of various models for years 2020,2021 as the testing data.
* To observe the impact of technical indicators on stock market prediction.
* To generate and visualize explanations for model predictions.

# Significance of the study

Stock market forecasting has always been a difficult job to accomplish. The random walk aspect of the stock market time series, as well as the large number of variables involved, are the primary reasons for this. However, studies have shown that historical data reflects stock prices, making movement patterns relevant for forecasting (Akhter and Mısır, 2005). Multiple studies have suggested that classical machine learning models either become too slow or too in-efficient for long range dependencies. Although neural network models like RNN, LSTMs have known to be more effective in context of stock market, their results are not interpretable. The significance of this study would be as follows:

* This study will compare the effectiveness of the novel Prophet model against the state-of-art time series prediction models for stock market prediction.
* This study will provide a comparison between the efficiency of the AR-Net neural network model against the state-of-art models for time series prediction and analyze its interpretability.
* This research will study and compare the impact of technical indicators on Prophet, AR-Net, and the state-of-the-art models.
* This study will evaluate the performance of time series prediction algorithms against the data from year 2020-2021.

# Scope of the study

The scope of this study is limited to following points:

* This study will use daily frequency data for fifteen stocks and sector indices for five different sectors.
* Data is based on trades done on Bombay stock exchange.
* This study is limited to technical analysis of the stock.
* This study makes use of up to six derived technical indicators as external variables for model building in addition to features available with the dataset.
* The impact of data transformation techniques is only investigated without the exogenous variables.
* The study will compare the performance of different models for the given dataset.

This study does not cover the following:

* Any trades done on NSE are not part of the dataset feature calculations.
* No macro-economic events or factors are taken into consideration.
* Financial information for the company is not provided as input to the models.
* News/special events specific to the stock are not included in the input parameters.

# Research Methodology

### Introduction

The main aim of the research is to establish the performance of novel time series prediction algorithms against the state-of-art machine learning algorithms and to study the impact of technical indicators on these models. The research is therefore divided into 2 major parts.

* Comparing the performance of the algorithms without any exogenous variables.
* Comparing the performance of the algorithms with the technical indicators.

Figure 1 represents the high-level approach proposed for this research.

Diagram

Description automatically generated

Figure 8.1: Research Methodology

### Dataset Description

The data has been sourced from quadl.com. Quandl provides a consolidated API to download stock market data from BSE. There are around 4500 stocks/indices listed on BSE. However, the stocks that are eligible for derivatives trading are generally used by traders. This allows the traders to use leverage and cash on the quick, short movements of the stocks. We use a combination of 15 stocks and sectorial indices from 5 different sectors for this study. The dataset consists of following attributes of each stock for every trading day for the period – January 1,2000 to March 18, 2021. Table 1 provides a description of features available in the dataset.

|  |  |  |
| --- | --- | --- |
| **Id** | **Feature** | **Description** |
|  | Date | Date of the trading day |
|  | Open | First traded price for the stock during the day. |
|  | High | Highest traded price for the stock during the day |
|  | Low | Lowest traded price for the stock during the day. |
|  | Close | Closing price for the day. |
|  | WAP | Weighted Average Price. |
|  | No. of Shares | Absolute number of shares traded for the day. |
|  | No. of Trades | Number of trades for the day for given script. |
|  | Total Turnover | Total value of the shares traded during the day. |
|  | Deliverable Quantity | Number of shares bought for delivery by traders during that day. |
|  | % Delivery Qty to Traded Qty | %age of traded quantity bought for delivery by investors |
|  | Spread H-L | Spread between the high and the low price for the day |
|  | Spread C-O | Spread between the opening and the closing prices for the day |
|  | Stock Name | Name of the listed company. |
|  | Code | Stock Code for Bombay stock exchange. |

Table 8.1:Features of the dataset

### Feature Engineering

This study derives technical indicators listed in Table 2 from the above data. The equations have been referred from (Financial Terms Dictionary, 2021)

|  |  |  |
| --- | --- | --- |
| **Id** | **Indicator** | **Calculation** |
|  | RSI |  |
|  | MACD |  |
|  | ATR |  |
|  | ADX | Calculation of ADX is a multi-step process referred from (Average Directional Index (ADX) Definition, 2021) |
|  | OBV | If    else |

Table 8.2: Technical Indicators

### Data pre-processing and transformations

Since we are using real time data from BSE, the initial dataset is clean. However, machine learning algorithms expect the time series to be in stationary format. We will perform 2 kinds of transformations on closing price of the stock before making predictions.

1. Change the closing prices to percentage returns format.
2. Log transformation.

The impact of these transformations will be studied on models without the exogenous variables.

In addition to this, after the feature extraction, it is inevitable to introduce null values. Also, different features may be on different scales. We will handle null values and perform relevant scaling and standardization techniques on the dataset.

* **Models**

In this study the focus will be on the state-of-art time series prediction models. As shown in the relevant research, Prophet is an additive model and Auto regressive neural network is an enhancement over prophet and ARIMA models. Therefore, this study will be using classical machine learning models for benchmarking the performance. The following listed models will be trained in the context of current study.

### ARMAX

Auto regressive model with exogenous variable is a generalization of ARIMA model which can incorporate external variables and model feedback. The ARMAX model is expressed as:

Where:

Yt: output series,

Xit: ith input time series or a difference of the ith input series at time t,

m: total number of external covariates,

ki: pure time delay for the effect of the ith input series,

ωi(B): numerator polynomial of the transfer function for the ith input series,

δi(B): denominator polynomial of the transfer function for the ith input series,

Nt: stochastic disturbance in the form of an ARMA model.(Li et al., 2014)

### SARIMAX

Seasonality in time series means when certain patters or behaviors occur at known intervals. To incorporate the impact of seasonality in certain businesses, we will use the Seasonal Autoregressive Integrated Moving Average Exogenous i.e., SARIMAX model. SARIMAX, like SARIMA applies ARIMA models to the lags that are integral multipliers of seasonality. Additionally, it takes exogenous variable as input. The SARIMAX (p, d, q) (P, D, Q, s) model is represented in the form of following equation-

### VARMAX

Vector auto regression moving average with exogenous regressors is an extension of VARMA model and hence a generalized form of ARMA. This model can be used for multivariate time series analysis and supports the use exogenous regressors. Like the ARMA model, this model is also characterized by ‘p’ and ‘q’ parameters. The VARMAX model is specified as:

### Prophet

Prophet forecasting model is designed to handle common features of business time series i.e., seasonality, cyclicity, outliers and impact of external variables. It is also designed to be easily interpretable to enable individual, who are not equipped with details of modelling, to be able to tune the model.(Taylor and Letham, 2017) Model proposed by prophet has 3 components – trend ), seasonality, holidays .

### AR-Net - Auto-Regressive Neural Network for time-series

AR-Net is a framework that is best of traditional models and neural networks. This model is proposed to work on time-series with long-range dependencies, needed for monitoring fine granularity data. This model overcomes the drawbacks of both AR Models and Neural networks. While dealing with long range dependencies, the AR models tend to become very slow. RNNs on the other hand are overly complex and lack interpretability. AR-Net is as interpretable as Classic-AR but also scales to long-range dependencies. (Triebe et al., 2019)

### Evaluation Metrics

We will use the following evaluation metrics for evaluating model performance.

### MAPE

Mean Absolute Percentage Error is a measure of prediction accuracy of the forecasting methods. It is given by the following equation:

Where: -

At: Actual value,

Ft: Forecast value.

### RMSE

Root mean squared error is a measure of the spread of the error. It is preferred evaluation metric as the scale and units match the dependent variable.

It is given by the following equation:

where:

xi: Actual observation,

x̂i: Predicted value,

N: Number of observations

### Adjusted R-squared

It measures how much of the variability of the dependent variables can be explained by the model. Adjusted R2 penalizes the model for excessive use of parameters. It is given by the following equations:

where:

p: number of predictors

N: total sample size

yi: actual observation

ŷi: predicted value

y̅: sample mean

### Noise

The noise metric calculates how much the model's predictions change from day to day. As you would expect, a model that changes its mind every few days is much more difficult to adopt (and much more costly to trade) than one that is more consistent.

### Model interpretability

Machine learning and deep learning methods are black box models. This study will assess the interpretability of the models using different methods such as LIME to explain the outcomes of the models. LIME can be extended to be used for time series data in addition to text, tabular or image data.

# Expected Outcomes

Following outcomes are expected from this research:

* To know the performance of novel time-series prediction methods versus the state-of-art algorithms.
* To observe the performance of novel and state-of-art algorithms for the stocks listed in Indian stock market.
* To assess the impact of technical indicators on different algorithms.

# Requirements / resources

This research is supported by the following Hardware and Software resources to carry out the experiments.

**Hardware Resources**

* Processor: Intel(R) Core (TM) i5-8250U CPU @ 1.60GHz, 1801 MHz, 4 Core(s), 8 Logical Processor(s)
* Memory: 16 GB RAM,
* Operating System: Windows 10 (64-bit operating system, x64 based processor)

For Neural Network training, this study will be using Nimblebox with below configurations.

* GPU - NVIDIA Tesla K80 12GB VRAM
* CPU - 4 Core
* RAM - 16GB LPDDR4
* HDD - 25GB

**Software Resources**

Following software and libraries will be required for performing the experiments.

1. Python (latest version)
2. Python Libraries
   * NumPy
   * Pandas
   * Matplotlib
   * pystan
   * NeuralProphet
   * Prophet
   * statsmodels
   * Lime-For-Time,
   * pycaret
3. IDE
   * Jupyter Notebook

# Research Plan

Chart

Description automatically generated with medium confidence

Figure 11.1: Research Plan

# Risk and Contingencies

|  |  |  |
| --- | --- | --- |
| **S. No** | **Perceived Risk** | **Mitigation Plan** |
|  | Number of model trainings required for this research may lead to requirements of additional computational resources. | 1. Use of freely available Google Colab accounts for parallel training, in addition to the available Nimble Box Resources.  2. Avoiding re-runs, whenever possible by saving the trained models. |
|  | Limitations of external libraries in supporting novel methods. | Buffer period has been considered in research plan estimation, for custom implementation or exploring additional existing libraries. |
|  | Unforeseen issue with system used for research. | Cloud storage providers like OneDrive and google drive will be used for regular back up of code and documentation.  Trained models will also be added to backup to allow restore. |
|  | Unexpected and unavoidable increased time demand from professional or personal commitments | Research will be done meticulously and adhering to all timelines. The research plan has been prepared to factor up to 5% extra time in case required in any phase. |