Decision Making: Applications in Management and Engineering
Vol. 4, Issue 1, 2021, pp. 51-86
ISSN: 2560-6018
eISSN: 2620-0104
DOI: https://doi.org/10.31181/dmame2104051g

FEB-STACKING AND FEB-DNN MODELS FOR STOCK TREND PREDICTION: A PERFORMANCE ANALYSIS FOR PRE AND POST COVID-19 PERIODS

Indranil Ghosh[1*] and Tamal Datta Chaudhuri[1]

[1] Calcutta Business School, West Bengal, India

Received: 28 October 2020;
Accepted: 21 January 2021;
Available online: 6 February 2021.

Original scientific paper

Abstract: *In this paper, stock price prediction is perceived as a binary classification problem where the goal is to predict whether an increase or decrease in closing prices is going to be observed the next day. The framework will be of use for both investors and traders. In the aftermath of the Covid-19 pandemic, global financial markets have seen growing uncertainty and volatility and as a consequence, precise prediction of stock price trend has emerged to be extremely challenging. In this background, we propose two integrated frameworks wherein rigorous feature engineering, methodology to sort out class imbalance, and predictive modeling are clubbed together to perform stock trend prediction during normal and new normal times. A number of technical and macroeconomic indicators are chosen as explanatory variables, which are further refined through dedicated feature engineering process by applying Kernel Principal Component (KPCA) analysis. Bootstrapping procedure has been used to deal with class imbalance. Finally, two separate Artificial Intelligence models namely, Stacking and Deep Neural Network models are deployed separately on feature engineered and bootstrapped samples for estimating trends in prices of underlying stocks during pre and post Covid-19 periods. Rigorous performance analysis and comparative evaluation with other well-known models justify the effectiveness and superiority of the proposed frameworks.*

Key words: *Binary Classification, Kernel Principal Component (KPCA), Bootstrapping, Stacking, Deep Neural Network.*

1. Introduction

The financial literature is replete with attempts in predicting stock prices. In contrast to the Efficient Market Hypothesis, researchers have identified various factors that can influence stock returns and hence have used them for prediction

* Corresponding author.
E-mail addresses: fri.indra@gmail.com (I. Ghosh), tamal5302@yahoo.com (T. Datta Chaudhuri)

purposes. Going back to Graham and Dodd (1934) where they disregarded the fact that "good stocks (or blue chips) were sound investments regardless of the price paid for them", they distinguished between speculation and investment, and consequently emphasized on factors like management quality, earnings, dividends, capital structure and interest cover. While econometric techniques have been predominantly used to predict stock returns, various machine learning tools like Artificial Neural Network, Support Vector Machine, Decision Tree, etc. have also been used for the purpose.

The literature can be classified according to choice of variables and techniques of estimation and forecasting. To mention a few, the first strand consists of studies using simple regression techniques on cross sectional data. Papers by of Basu (1977, 1983), Jaffe et al. (1989), Banz (1981), Fama and French (1988, 1992, 1995), Strong and Xu (1997), and Ibbotson and Idzorek (1998) fall into this category.

The second strand of the literature uses time series models and techniques to forecast stock returns. Some papers in this area are by Srinivasan and Prakasam (2014), Babu and Reddy (2015) and Ahmar and Val (2020). Econometric tools like autoregressive integrated moving average (ARIMA), autoregressive distributed lag (ARDL), generalized autoregressive conditional heteroscedasticity (GARCH) have been employed to forecast stock prices.

Papers by Mostafa (2010), Dutta et al. (2006), Shen et al. (2007), Chen et al. (2003), Wu et al. (2008), Perez-Rodriguez et al. (2005) and Datta Chaudhuri et al. (2016, 2017), Ghosh et al. (2018) fall in a third category where machine learning tools have been used for prediction of stock returns. Majority of these studies applied traditional or variants of artificial intelligence driven (AI) models for prediction of stock returns. Sezer et al. (2020) conducted an exhaustive and systematic review of usage of deep learning driven models for financial time series forecasting. Their work illustrates the usage of deep neural network (DNN), recurrent neural network (RNN), long short-term memory network (LSTM), convolutional neural network (CNN), restricted Boltzmann machine (RBM) method, deep belief network (DBN), auto encoder (AE), and deep reinforcement learning (DRL) on plethora of equity market data. Work of Jiang et al. (2020) also presents a review of applications of deep learning models, features, and deployment text and image data for stock market data. The study outlines effectiveness of additional deep learning models, graph neural network (GNN), gated recurrent unit (GRU) and discriminative deep neural network with hierarchical attention (HAN) for forecasting. Usage of technical indicators and feature engineering through principal component analysis (PCA) has been reported as well. Rundo et al. (2019) thoroughly reviewed frameworks using econometric methods, machine learning, and deep learning methods for predictive modelling of Asian, European, and US stock markets. Their study also covered the indices commonly used for evaluating models. Amongst the machine learning models, support vector machine (SVM), decision tree (DT), random forest (RF), boosting, and artificial neural network (ANN) have also been successful in modelling financial markets. Therefore, in our paper, utilizing deep learning and machine learning frameworks in an integrated framework for predictive analysis, is justified.

Prediction of stock price movements is critical for stock market traders and portfolio managers as they have to continuously realign their strategies with market volatility. Recent times have observed increase in research on stock price prediction based on advanced AI based frameworks. The stock market prediction problem can broadly be categorized into two strands. The first category deals with estimation of closing prices of different stocks, while the second strand attempts to predict the direction of movement i.e. whether stock prices would increase or decrease after a pre-specified time interval. The second category of problem is also referred as

classification problem. The problem is quite challenging as correct estimation of trend can immensely boost investors for trading as compared to buy and hold strategy for long duration. Additionally imbalanced distribution of class information of target variable, known as class imbalance often further complicates the task (Pirizadeh et al. 2020, Bria et al. 2020) which may lead to poor performance in test data cases. Predominantly several variations of sampling strategies are used to tackle the problem (Shin et al. 2021). Our research attempts to develop an integrated research structure capable of modeling class imbalance in order to carry out stock trend classification in Indian context.

The body of research mentioned above has focused on relatively low volatile and chaotic time horizons. However, the outbreak of Covid-19 pandemic has wreaked havoc by disrupting business and global supply chains. To curb infections, nations across the world resorted to strict lockdowns, banned international travels, sealed borders and imposed restriction on movements of goods and people which eventually led to increased uncertainty and stock market volatility. It would be interesting and important to check whether stock price trends can be predicted with some degree of accuracy during the new normal period owing to Covid-19 pandemic. It also needs to be seen whether AI driven frameworks can be useful in such situations.

One step ahead stock price trend prediction is a process of foretelling whether price of the underlying stock would increase or decrease. An increase would indicate buy signal (up) while decrease would reflect sell signal (down). Hence the problem basically takes the form of binary classification. The said problem is often affected by class imbalance, i.e. disproportion between buy and sell ratio. It is highly probable to have class imbalance during the Covid-19 period. Considering these challenges, it becomes absolutely imperative to design robust frameworks for predictive modeling of stock price trends and test the same in new normal time periods.

In this paper, we have considered four Indian companies namely, HDFC Bank, Tata Consultancy Services (TCS), Reliance Industries Ltd. (RELIANCE), and Spice Jet Limited (SPICEJET) as examples for predicting their future stock price trends. They belong to four different sectors namely, banking, IT, energy and airlines. These companies have been consistently profit making and dividend paying, are leaders in their respective sectors in terms of size and performance and their stocks are extensively traded in the Indian stock market. Among the four sectors, airlines sector has been a recipient of rapid shock owing to worldwide lockdown due to Covid pandemic. Thus, our framework would be tested for efficacy on challenging time series data as well. The interested reader can consider other companies and test the efficacy of our framework.

This paper considers technical indicators along with macroeconomic variables as explanatory variables for predicting the trend of aforesaid stocks. The exercise has been carried out on different time frames covering pre-Covid-19 and Covid-19 periods. Rigorous feature engineering (FE) process has been evoked using unsupervised feature selection algorithm i.e., kernel principal component analysis (KPCA) for better realization and compactness of dataset in high dimensional feature space. The class imbalance obstacle has been resolved through bootstrapping process. Both FE and bootstrapping processes are invoked before applying AI algorithms for discovering the association between the explanatory and target variables for precisely predicting the trend. Models belonging to two sub-fields of AI, machine learning and deep learning have been exploited for the predicting exercise. Stacking, a machine learning framework built upon combination of various other learning algorithms for classification, has also been used for predicting the price trend of the three stocks. The

stacking architecture has been built by combining three ensemble machine learning algorithms namely, random forest (RF), bagging, and gradient boosting (GB).

Since stacking is driven by both FE and bootstrapping operations, the combined framework has been coined as FEB-Stacking. Deep neural network has been utilized for predicting trends. Like the FEB-Stacking approach, DNN has been deployed in conjunction with FE and bootstrapping processes. Hence, the combined framework has been referred as FEB-DNN. Rigorous classification accuracy measures have been computed to ascertain the predictive accuracy of both FEB-Stacking and FEB-DNN models. Profitability of both frameworks has been compared against the profitability of buy and hold strategy. Further, comparative study with several benchmark models has been conducted to properly justify the use of the proposed architectures.

The major contribution of the present research work lies in designing predictive structures in challenging times like Covid-19 where financial markets are highly volatile and when financial markets experience crashes in stock market and worldwide recession. The paper proposes a structured framework for selecting technical and macroeconomic indicators for building the trend prediction frameworks. Our approach recognizes the class imbalance problem arising in volatile times and combining such processes with stacking and DNN models and checking the effectiveness in Covid-19 pandemic time horizons comprise the novelty of our work. Both FEB-Stacking and FEB-DNN are exposed to a battery of performance tests to prove the efficiency.

The remaining portion of the article is organized as follows. Section 2 outlines the previous related research to comprehend the evolution pattern and identify the existing gaps. Subsequently, brief description of the data for accomplishing our research endeavor is provided in Section 3. The entire working principle and the research methodology is then elucidated in Section 4. Next, predictive results are presented in detail and discussed in Section 5. Section 6 concludes the paper highlighting the key implications and future research potential.

2. Previous Research

Stock price predictive modeling has garnered strong focus among researchers and practitioners owing to its practical implications and arduous nature of modeling. As stated earlier, the predictive modeling of financial markets can be categorized in two strands namely, forecasting absolute figures and estimating trend direction. Plethora of AI driven models have been reported to be extremely successful in capturing inherent and complex pattern driving stock market dynamics. It should also be noted that research aiming at predictive analysis has not been restricted to stock market time series data only. Other financial time series variables viz. volatility, exchange rate and commodity prices too have been explored for forecasting exercises.

Atsalakis and Valavanis (2009) developed a predictive structure based on adaptive neuro fuzzy inference system (ANFIS) for forecasting returns of stock markets of Athens and New York. The model emerged to yield forecasts of supreme accuracy and more profitable than buy and hold (B&H) strategy.

Zhang et al. (2016) developed a hybrid technical indicator driven stock trend prediction system comprising adaboost, probabilistic support vector machine (PSVM) and genetic algorithm. PSVM was used as base learner in adaboost while GA assisted in optimal hyper-parameter tuning. Rigorous performance inspection demonstrated the classification accuracy and trading benefits.

Chatzis et al. (2018) conducted predictive modeling exercises of global stock, bond, and currency markets using a series of machine learning and deep learning models during the time horizons affected by several stock market crash events. They mainly utilized salient fundamental features pertinent to respective market as explanatory features which were evaluated using Boruta feature selection algorithm. As predictive modeler, Classification Trees, Support Vector Machines, Random Forests, Neural Networks, Extreme Gradient Boosting, and Deep Neural Networks were used. Findings revealed insights of practical relevance.

Chen and Hao (2018) proposed a stock trading signal prediction system incorporating PCA and weighted SVM. PCA was used on raw technical indicators for refinement and feature engineering process. The transformed feature set was used in weighted SVM model for prediction performance. Efficacy of the proposed model was validated on Shanghai and Shenzhen stock markets.

Lei (2018) developed a framework for stock price trend prediction using hybrid framework of rough set (RS) and wavelet neural network (WNN). The framework utilized several technical indicators as explanatory features which were refined through RS based feature selection model. Subsequently WNN was trained on selected feature set for performing predictive exercise. Efficacy of developed model was validated on trend estimation of SSE Composite Index, CSI 300 Index, All Ordinaries Index, Nikkei 225 Index and Dow Jones Index.

Bisoi et al. (2019) developed a hybrid granular predictive structure comprising variational mode decomposition (VMD), differential evolution (DE), and a robust kernel extreme learning machine (RKELM) technique for forecasting daily prices of BSE S&P 500 Index (BSE), Hang Seng Index (HSI) and Financial Times Stock Exchange 100 Index (FTSE). VMD was deployed to better model the inherent nonlinearity, DE was used for optimal parameter tuning while final prediction were drawn using RKLM. The framework emerged superior to several well-known algorithms.

Das et al. (2019) developed an integrated model of feature selection and predictive modeling of BSE Sensex, NSE Sensex, S&P 500 index and FTSE index. Hybrid structure of principal component analysis (PCA) and several metaheuristic searching algorithms, firefly optimization (FO) and GA was utilized for feature engineering on a set of technical indicators. Subsequently, machine learning algorithms, extreme learning machine (ELM), online sequential extreme learning machine (OSELM) and recurrent back propagation neural network (RBPNN) were used for estimating forecasts on different time intervals. Among these methods, OSELM appeared to be superior.

Zhou et al. (2019) proposed a hybrid predictive framework of empirical mode decomposition (EMD) and factorization machine based neural network for daily closing price prediction of Shanghai Stock Exchange Composite (SSEC) Index, the National Association of Securities Dealers Automated Quotations (NASDAQ) Index and the Standard & Poor's 500 Composite Stock Price Index (S&P 500). The predictive performance duly rationalized the efficiency of proposed architecture.

Ismail et al. (2020) developed a feature engineering structure based on persistent homology to form more meaningful explanatory features from original feature set for trend prediction of Kuala Lumpur stock exchange. The outcome of persistent homology was fed into logistic regression, artificial neural network, support vector machine and random forest for estimating one day-ahead trend movement. The combination of persistent homology and SVM emerged to be the most efficient one.

Liu and Long (2020) proposed a novel deep learning framework for stock market prediction. The framework utilized empirical wavelet transform (EWT) and outlier robust extreme learning machine (ORELM) for preprocessing and long short term

memory network (LSTM) for forecasting. Further fine tuning of LSTM was carried out using particle swarm optimization (PSO). The framework emerged to be superior to several benchmark models.

Carta et al. (2021) developed a reinforcement learning framework based on ensemble of deep Q learning agents for predictive analysis of stock markets. Unlike machine and deep learning models, the reinforcement learning strategy was implemented by training Q-learning agent on same training samples. The framework emerged to yield excellent trading performance in comparison to conventional B&H strategy.

Review of the existing literature clearly indicates extensive usage of machine and deep learning driven models in stock market forecasting and classification. Clear trend of hybrid granular models incorporating such models is also apparent. Recently, stock market sentiment analysis and reinforcement learning have appeared to significantly contribute to precise modeling of stock market trends and absolute figures too. Methodologically, either technical indicators or macro-economic variables have been predominantly used as explanatory features. Nevertheless, frameworks built on amalgamation of both types of features to carry out predictive exercises in extreme volatile regimes are absent. On the other hand, behavior, co-movement, causality of various stock markets during the global financial crisis have received serious attention in the literature. Characterization of stock market crashes have been elaborated as well. However, development of predictive frameworks to estimate trends during unprecedented or black swan events has seen comparatively less attention. Specifically, there is paucity of predictive models to estimate financial market trend during Covid-19 pandemic. Moreover, the task of trend modeling needs to properly combat class imbalance and proper feature engineering issues. Therefore, design of integrated frameworks to yield precise forecasts for severe conditions is of paramount significance. Our research attempts to address these challenges and endeavor to design a robust framework which can significantly contribute to the previous literature on stock market prediction.

3. Data and Variable Description

3.1. Data

To accomplish the research objectives, we have compiled daily closing prices of HDFC Bank, TCS, RELIANCE, and SPICEJET from January, 2014 to July, 2020. For performing stock price trend prediction, the datasets are segregated into two strands reflecting different time horizons. The first set comprises of data ranging from January, 2014 to December, 2019 which has been referred as Set A throughout the paper. On the other hand closing price data of underlying stocks from January, 2014 to July, 2020 forms Set B. The partitioning has been made in order to assess the classification accuracy of proposed predictive models on relatively less volatile time horizons and on time horizons deeply penetrated by the impact of Covid-19 pandemic. Thus, any analysis on Set A would measure the effectiveness of proposed frameworks in trend estimation during pre-Covid time horizons, whereas analysis with Set B would measure quality of predictions during post-Covid time horizon. Figures 1 and 2 exhibit the evolutionary pattern of temporal movements of underlying variables.

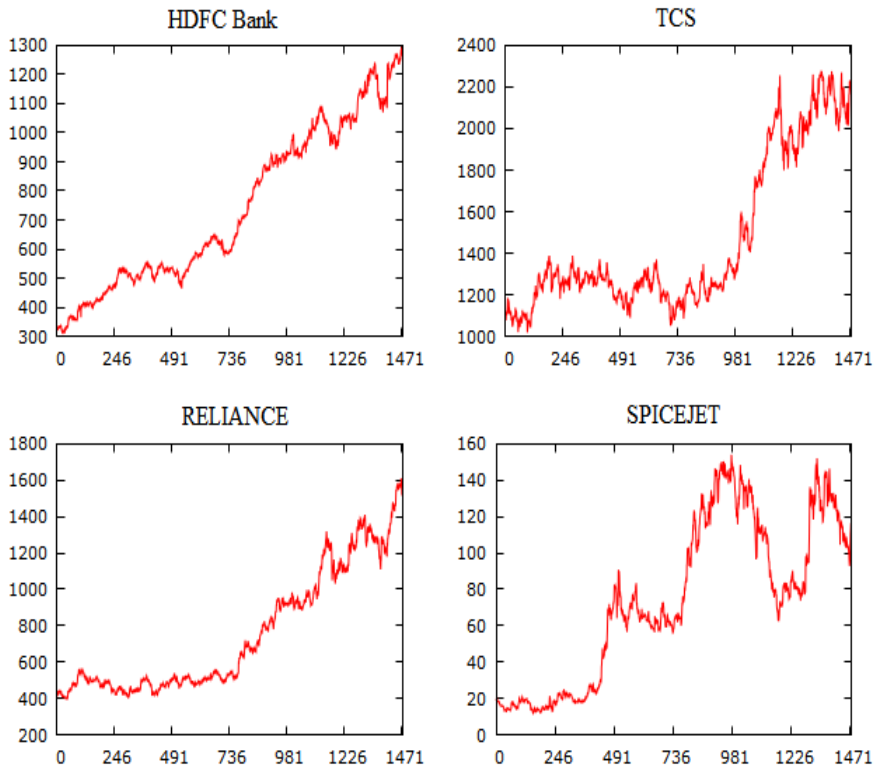

Figure 1. Temporal Evolutionary Movements of Set A Dataset

During the pre-Covid context, i.e. Set A dataset, it can be observed that HDFC Bank and RELIANCE stock prices more or less exhibit dominance of trend component over short term fluctuations. TCS stock prices on the other hand demonstrate comparatively more fluctuation in addition to trend component. Finally, SPICEJET stock prices exhibit periodic pattern with growth. Hence, outcome of visual inspection suggests that Banking and Energy sector have performed reasonably well, while performance of IT sector has undergone certain extent of uncertainty during the said time horizon. The figure of the stock price movement of the airline company reflects seasonality.

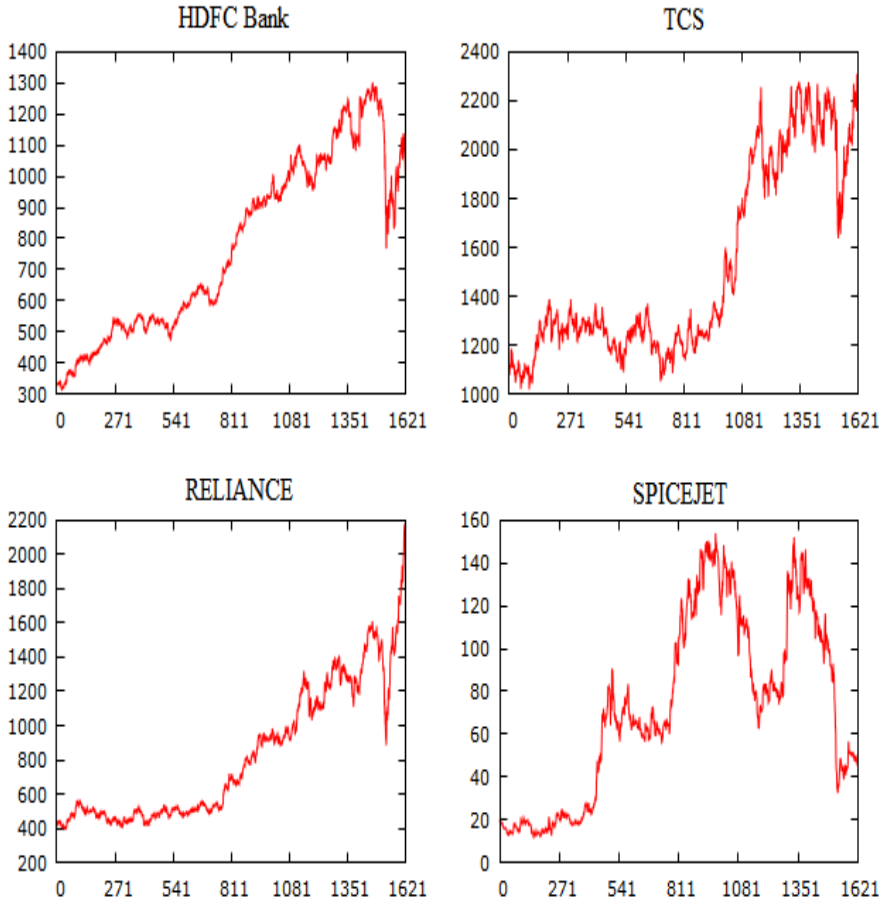

Figure 2. Temporal Evolutionary Movements of Set B Dataset

Visualization of Set B dataset, reflecting the impact of Covid fear, clearly demonstrates drastic falls in the stock prices of selected companies. Of late, stock prices of HDFC Bank, TCS and RELIANCE have displayed signs of recovery. The stock prices of SPICEJET, however, have not recovered from the Covid shock as there exist curbs on airline movements to varying extent till now.

Briefly speaking, the selection of the sectors as well as the segregation of the time horizons, make the forecasting task extremely challenging and arduous. For better understanding of critical properties, descriptive statistics have been computed as well. Tables 1 and 2 outline key statistical properties of the datasets.

Table 1. Descriptive Statistics of Set A Dataset

Properties	**HDFC Bank**	**TCS**	**RELIANCE**	**SPICEJET**
Minimum	313.2	1018	400.0	11.25
Maximum	1302.4	2278	1610.0	154.00
Mean	747.2	1464	757.0	74.15
Median	642.5	1280	541.2	71.55
SD	277.630	370.58	336.288	44.219
Skewness	0.302	0.938	0.76	0.057
Kurtosis	-1.284	-0.734	-0.824	-1.312
Jarque-Bera	123.66***	249.96***	184.06***	10.6.25***
Shapiro Wilk	0.9216***	0.797***	0.841***	0.914***
Frosini Test	2.3723***	4.139***	3.164***	1.713***
ADF Test	2.5072#	1.0334#	1.9645#	0.0794#
Terasvirta's NN Test	42.92***	19.459***	11.885#	9.8803***
Hurst Exponent	0.8918	0.8844	0.8886	0.8813

***Significant at 1% level of significance, #Not Significant, SD: Standard Deviation, ADF: Augmented Dickey Fuller, NN: Neural Network

Table 2. Descriptive Statistics of Set B Dataset

Properties	**HDFC Bank**	**TCS**	**RELIANCE**	**SPICEJET**
Minimum	313.2	1018	400.0	11.25
Maximum	1302.4	2310	2177.7	154.00
Mean	775.2	1515	823.5	72.89
Median	732.1	1296	663.9	69.38
SD	282.698	393.257	392.914	42.907
Skewness	0.1582	0.670	0.772	0.128
Kurtosis	-1.3590	-1.218	-0.470	-1.242
Jarque-Bera Test	131.4***	221.68***	176.18***	108.31***
Shapiro Wilk Test	0.9269***	0.822***	0.863***	0.927***
Frosini Test	2.3531***	3.985***	2.963***	1.5969***
ADF Test	0.8139#	0.9364#	2.4592#	-0.5647#
Terasvirta's NN Test	25.522***	52.076***	32.811***	6.2511**
Hurst Exponent	0.8936	0.8928	0.8888	0.8828

***Significant at 1% level of significance, #Not Significant, SD: Standard Deviation, ADF: Augmented Dickey Fuller, NN: Neural Network

It is evident that none of the underlying stocks follow normal distribution while presence of non-stationary evolutionary pattern is also apparent as manifested by outcome of Jarque-Bera, Frosini, and Shapiro-Wilk tests. Results of ADF test clearly indicates selected stock prices are non-stationary in nature. Outcome of nonlinearity assessment through Terasvirta's neural network test suggests entrenchment of nonlinear traits in all four stocks for Set B datasets considering Covid-19 period. In Set A segment reflecting normal time horizon, TCS, RELIANCE, and SPICEJET stock prices

have emerged to be nonlinear. On the flipside, estimated Hurst exponent figures imply the underlying time series observations of both sets exhibit long memory dependence or persistent pattern as the they are substantially greater than 0.5 (Ghosh and Datta Chaudhuri, 2018). Successful usage of technical indicators for predictive modeling of financial time series observations exhibiting persistent pattern has been reported in literature. Therefore integration of technical indicators for trend prediction of chosen stocks is justified. Since high degree of non-stationary and nonlinear traits with complete nonparametric movements can be observed, deployment of advanced AI models is considered appropriate.

3.2. Variables

The present work is aimed at stock trend prediction, i.e. to estimate whether one-day ahead closing price would increase or decrease. An increase would indicate an 'up' signal while decrease refers 'down' signal. Thus objective of proposed research methodology is to correctly classify the next day movement. The aforesaid problem is also referred as binary classification the target takes two classes explicitly. Mathematically the target (T) can be explained as:

$$T = \begin{cases} 0 \; if \; (P_i - P_{i-1}) < 0 \\ 1 \; if \; (P_i - P_{i-1}) \geq 0 \end{cases} \quad (1)$$

Where, P_i and P_{i-1} represent closing prices of two consecutive days of any stock

We attempt to develop a robust predictive structure to estimate the future trend direction, i.e. 0 ('down') and 1 ('up') of HDFC Bank, TCS, RELIANCE, and SPICEJET share prices. As empirical analysis of considered datasets hint at existence of long memory dependence, several technical indicators which are computed by performing simple mathematical operations on closing prices have been selected as explanatory features as outlined in Table 3.

Table 3. List of Technical Indicators

No	Feature	Formulae
1.	One day back closing price (LAG1)	$LAG1 = P_{i-1}$ where P_{i-1} denotes closing value at previous day
2.	Two-day back closing price (LAG2)	$LAG2 = P_{i-2}$
3.	Three-day back closing price (LAG3)	$LAG3 = P_{i-3}$
4.	Four-day back closing price (LAG4)	$LAG4 = P_{i-4}$
5.	Five-day back closing price (LAG5)	$LAG5 = P_{i-5}$
6.	5-day moving average (MA5)	$MA5 = \frac{\sum_{i=j-4}^{j} P_i}{5}$
7.	10-day moving average (MA10)	$MA10 = \frac{\sum_{i=j-9}^{j} P_i}{10}$
8	20-day moving average (MA20)	$MA20 = \frac{\sum_{i=j-19}^{j} P_i}{20}$

No	Feature	Formulae
9.	5-day bias (B5)	$B5 = \frac{P_i - MA5}{MA5}$
10.	10-day bias (B10)	$B10 = \frac{P_i - MA10}{MA10}$
11.	20-day bias (B20)	$B200 = \frac{P_i - MA20}{MA20}$
12.	5-day momentum (MTM5)	$MTM5 = P_i - P_{i-5}$
13.	10-day momentum (MTM10)	$MTM10 = P_i - P_{i-10}$
14.	20-day momentum (MTM20)	$MTM20 = P_i - P_{i-20}$
15.	5-day exponential moving average (EMA5)	$EMA5 = \frac{2}{5+1} \times P_5 + \frac{5-1}{5+1} \times EMA4$, where $EMA1 = P_1$
16.	10-day exponential moving average (EMA10)	$EMA10 = \frac{2}{10+1} \times P_9 + \frac{10-1}{10+1} \times EMA9$
17.	20-day exponential moving average (EMA10)	$EMA20 = \frac{2}{20+1} \times P_{19} + \frac{20-1}{20+1} \times EMA19$
18.	5-day rate of change (ROC5)	$ROC5 = \frac{P_i - P_{i-5}}{P_{i-5}}$
19.	10-day rate of change (ROC10)	$ROC10 = \frac{P_i - P_{i-10}}{EC_{i-10}}$
20.	20-day rate of change (ROC20)	$ROC20 = \frac{P_i - P_{i-20}}{EC_{i-20}}$
21.	Upper Bollinger band (UB)	$UB = MA20 + (20 \times \sigma_{20})$ where σ_{20} denotes standard deviation of previous 20 days closing prices
22.	Lower Bollinger band (LB)	$LB = MA20 - (20 \times \sigma_{20})$
23.	Difference (DIFF)	$DIFF = EMA26 - EMA12$
24.	Moving Average Convergence Divergence (MACD)	$MACD = 2 \times (DIFF - DEA)$; $DEA = EMA(DIFF)$
25.	Difference of High and Low Price (H-L)	$H - L = HP_{i-1} - LP_{i-1}$; HP_{i-1} and LP_{i-1} denote high and low price of previous day
26.	Difference of Closing and Opening Price (C-O)	$C - O = CP_{i-1} - OP_{i-1}$; CP_{i-1} and OP_{i-1} denote closing and opening price of previous day

Alongside technical indicators, several key macroeconomic features representing sector outlook, raw material prices, market fear, and market sentiment have been added to the explanatory variable list as well. As discussed earlier, majority of past

literature have either relied either on technical features or on macroeconomic constructs. The present paper combines them to achieve better classification accuracy in extreme circumstances. Table 4 reports the macroeconomic variables used in the analysis.

Table 4. Macroeconomic Variable Details

Stocks	Macroeconomic Indicators	Components
HDFC Bank	NIFTY, INDIA VIX, NIFTY Bank Index	Market Sentiment, Market Fear, and Sectoral Outlook
TCS	NIFTY, INDIA VIX, IT Sectoral Index, Rupee-Dollar exchange rate	Market Sentiment, Market Fear, Sectoral Outlook, and Foreign Exchange Rate
RELIANCE	NIFTY, INDIA VIX, ENERGY Sectoral Index, Crude Oil Price	Market Sentiment, Market Fear, Sectoral Outlook, and Raw Material Price.
SPICEJET	NIFTY, INDIA VIX, Crude Oil Price	Market Sentiment, Market Fear, and Raw Material Price.

Technical features remain uniform for all four stocks while the macroeconomic features vary according to the industry segment. The combined set of raw explanatory features will undergo rigorous feature engineering process through KPCA technique before being deployed for the prediction process.

4. Methodology

This section articulates the utilized components of integrated predictive architectures, FEB-Stacking and FEB-DNN chronologically. Figure 3 depicts the integrated research framework.

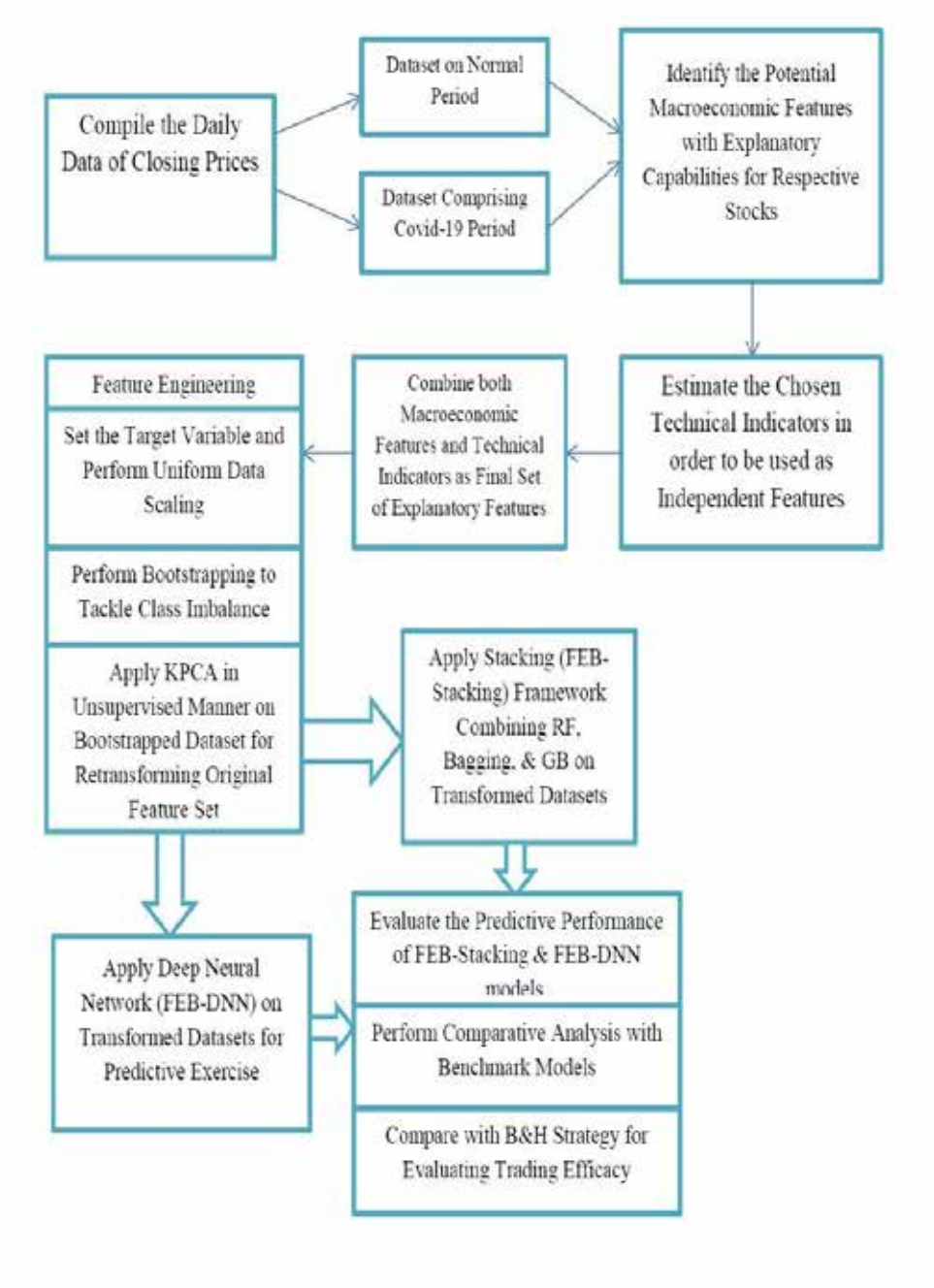

Figure 3. Flowchart of Research Framework

The structure of the integrated research model shown in Figure 3 demonstrates the flow of deployment of the different components in a seamless manner. Initially after compilation and segregation of datasets across pre-Covid and post-Covid regimes, macroeconomic indicators and technical features are arranged as explanatory variables for estimating trend of chosen stocks. Subsequently, bootstrapping and KPCA have been evoked to sort class imbalance problem and

feature engineering process respectively. Stacking and DNN models are then applied on feature engineered and bootstrapped samples for carrying out predictive analysis to automatically estimate trends. A battery of numerical evaluations and statistical tests are utilized to critically assess effectiveness of both forecasting frameworks. We next, briefly expound the principles of utilized research components.

4.1. Kernel Principal Component Analysis (KPCA)

Kernel Principal Component Analysis (KPCA) is an extension of ordinary PCA method (Scholkopf et al., 1999), where to tackle non linearity, input data space is mapped into feature space. Usually a kernel function is used to carry out the inner products in the feature space without explicitly defining transformation φ. The present research utilizes well known radial basis kernel for accomplishing the task. After the said transformation, orthodox PCA is invoked on the transformed dataset. Applying KPCA transformation, we project the raw set of explanatory features comprising technical and macroeconomic indicators into feature space which would be ideal for precise classification of the futuristic trends of chosen stocks. Thus, the objective of FE process through KPCA is not to reduce feature set but to obtain retransformation for better predictions. We next explain how class imbalance problem has been tackled in the proposed predictive architectures.

4.2. Fixing Class Imbalance

The problem of data classification refers to the imbalance distribution of target variable classes in the dataset. The target variable which we have set in this study is strictly binary in nature. However, anticipation of crash in markets, uneven bearish and bullish phases may lead into severe imbalance in distribution of the target construct. There exists high possibility for models built on such dataset to exhibit over-fitting phenomenon, thereby performing poorly in test data segments. Thus, it is necessary to balance the ratio of up and down signals of our dataset to be balanced beforehand. Literature reports usage of random up and down resampling approaches as bootstrapping driven solution for dealing with class imbalance problem. In this work, we have opted for up-sampling approach to generate artificial data in order to compensate the lagging proportion of a particular class depending on actual count. The ratio of 'up' (1) and 'down' (0) signals as expressed by equation 1 is estimated beforehand and up-sampling is applied to the lagging signals in order to keep the ratio even.

We now proceed to discuss the principles of stacking and DNN used for yielding predictions exhaustively.

4.3. Stacking

It replicates the working principle of typical ensemble machine learning frameworks where predictions from multiple models are used as inputs to yield the final predictions for developing forecasting framework. In this work, stacking has been applied on predictions obtained through three different ensemble learning models namely, gradient boosting (GB), random forest (RF), and bagging. The final training of stacking is achieved through deploying a separate RF model, with 200 base learners, which acts as final stacking classifier. Detailed of constituent models have been elucidated as follows. The stacking framework has been implemented using ensemble utilities of *'sklearn'* library of Python.

4.3.1. Gradient Boosting (GB)

Boosting is an ensemble predictive analysis technique where a series of different learning algorithms are applied in a forward-stage wise manner to generate final predictions (Schapire and Singer, 1999). Gradient boosting is a variant of classical boosting algorithm which basically mimics the same principle with an extension of identification of training samples via determination gradient driven error rate computation. Decision trees for classification have been used as base learners sequentially in forwards direction. Simulation of the method has been carried out using *'sklearn'* library in python programming environment. In the implementation part of GB algorithm, learning rate (0.05), number of base learners (300), maximum number of feature (7), and maximum depth (5) of decision trees have been considered for hyper parameter tuning which is basically accomplished through *'GridSearch'* utility of Python library. Default figures of other parameters have been considered.

4.3.2. Random Forest (RF)

It is an ensemble based machine learning model comprising decision trees as base learners. RF, developed by Breiman (2001), is characterized by its high precision, robustness to outliers and effective execution time. Since inception, it has garnered tremendous attention among the academic fraternity and practitioners for solving classification and regression tasks (Lariviere and Van den Poel, 2005; Liu et al., 2013). Since the underlying research problem of the paper is binary classification, decision trees for classification have been chosen as base learners. Number of base learners in RF can be arbitrary and depend on complexity of the problem. Final assignment of class label information (for classification task) or estimation of continuous outcome (for regression task) on test data set is carried out through majority voting or averaging scheme. Three parameters namely, maximum features (8), number of base learners (500), and minimum number of samples for split (2), have been fine-tuned using *'GridSearch'* utility of Python library, while default values of other parameters have been considered.

4.3.3. Bagging

Similar to RF, bagging (Bootstrap Aggregating) also follows similar ensemble properties for modelling data classification tasks (Lemmens and Croux, 2006; Zheng et al. 2011; Simidjievski et al., 2015). It too utilizes decision tree for classification as constituent base learner. Majority voting scheme is applied to draw final predictions based on the outcome of individual trees which grow in bootstrapped samples drawn from training samples. Outcome of individual clan ensemble based predictive modelling technique however, differs from former in implementation ensemble learning. Bagging reduces the variance of unstable learning methods leading to improved prediction. There are differences between Bagging and RF. Only a subset of features are chosen randomly from set of all features for splitting operations of constituent decision trees in RF, whereas Bagging evaluates all features to identify the most suitable for splitting operations. Thus, incorporating RF and Bagging together in stacking structure would cancel out the effects of over fitting and under fitting. For implementing Bagging, number of base learners (350), maximum numbers of features (8), maximum samples (1.0), and feature bootstrapping (False) have been auto-tuned using *'GridSearch'* utility keeping default values of remaining parameters.

GB, RF, and Bagging receive technical indicators and macroeconomic indicators of respective stocks outlined in Tables 3 and 4 as inputs for predicting the target defined

in Equation 1. Predictions obtained by the three models are fed as inputs in the stacking framework to obtain the final predictions.

The entire modelling, i.e. combination of FE through KPCA, bootstrapping via up-sampling for sorting class imbalance, and Stacking for yielding prediction, has been implemented using Python programming language. As stated, stacking combines the outcome of GB, RF, and Bagging and treat them as new set of features for explaining the movements of trend. It should be noted that all these methods are dependent on several process hyper-parameters which have been auto tuned invoking 'GridSearch' utility of Python library. The integrated FEB-Stacking has been evaluated separately on Set A and Set B observations for ascertaining performance in pre-Covid and post-Covid time periods distinctly. For assessing the predictive performance, typical classification measures viz. ROC curve, specificity, sensitivity, and various other measures have been used as discussed in sub-sections 4.5 and 4.6 respectively.

4.4. Deep Neural Network (DNN)

Artificial neural network (ANN) models have emerged to be highly effective and successful in modeling complex pattern recognition problems throughout the literature. The ANN architecture comprises of three distinct layers, input layer, hidden layer, and output layer. With rapid development and success of deep learning methodologies, a subset of AI field, focus has been put to examine efficacy deep neural network (DNN) structures where multiple hidden layers are incorporated in standard ANN architecture for carrying out predictive analysis tasks (Liu et al., 2017; Qureshi et al., 2017). These hidden layers act as additional feature engineering process in the context of predictive modeling tasks. In this problem, these layers additionally refine the fed input features for performing classification. Individual hidden layers of DNN comprise of several neurons connected to neurons of adjacent layers. They receive inputs from the previous layer and estimate output for propagation to next layer. In this work two hidden layers of 50 nodes each, have been deployed. Transformation functions are utilized for generation of output through deployment of activation functions. Literature reports different activation functions including 'identity', 'sigmoid', 'tanh', and 'relu'. In this research 'relu' (rectified linear unit) function has been used as activation function. The training of DNN is achieved through adjusting connection weights and biases based on the amount of error in the output compared to the expected result encapsulated in the loss function. This learning process is carried out through forward- and back-propagation and solved by the "adam" optimizer, which is an algorithm for optimization of stochastic objective functions, proposed by Kingma and Ba (2014). All technical and macroeconomic indicators comprise the input layer, which undergoes series of transformations in hidden layer in order to generate the future trend as output.

Feature engineering and bootstrapping processes are combined with DNN to form FEB-DNN model to estimate trend predictions of HDFC Bank, TCS, RELIANCE, and SPICEJET during normal and new-normal time horizons. The model is simulated using Keras interface in Python programming framework. Likewise FEB-Stacking, Set A and Set B data samples are used to test predictive ability of FEB-DNN at pre-Covid and post-Covid time frames. To evaluate the classification performance of respective models, visual metric and quantitative indices have been obtained. Visual metric in the form of receiver operating characteristic (ROC) curve is determined while several quantitative binary classification indices are estimated also.

4.5. Receiver Operating Characteristic (ROC) Curve

It is used for evaluating the predictive performance of a classifier and seldom utilized for model selection. ROC curve depicts a visualization of sensitivity represented by vertical axis and 1-specificity represented by horizontal axis. Basically, it reflects the probability of correctly specifying a random pair of positive and negative instances. To get quantitative information from ROC curve, area under the curve (AUC) is estimated. Models associated with higher AUC values are said to yield better and accurate predictions. It should be close to 1 to indicate superior classification performance.

4.6. Quantitative Measures

To evaluate efficiency of proposed predictive structures, FEB-Stacking and FEB-DNN, the present research has utilized a series of quantitative indices which are mathematically expressed as:

$$Sensitivity = \frac{TP}{TP+FN} \quad (2)$$

$$Specificity = \frac{TN}{TN+FP} \quad (3)$$

$$G = \sqrt{Sensitivity * Specificity} \quad (4)$$

$$LP = \frac{Sensitivity}{1-Specificity} \quad (5)$$

$$LR = \frac{1-Sensitivity}{Specificity} \quad (6)$$

$$DP = \frac{\sqrt{3}}{\pi}\left[log\frac{Sensitivity}{1-Specificity} + log\frac{Specificity}{1-Sensitivity}\right] \quad (7)$$

$$\gamma = Sensitivity - (1 - Specificity) \quad (8)$$

$$BA = \frac{1}{2}(Sensitivity + Specificity) \quad (9)$$

$$MCC = \frac{(TP\times TN)-(FP\times FN)}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}} \quad (10)$$

$$F1 = \frac{2TP}{2TP+FP+FN} \quad (11)$$

$$FM = \sqrt{\frac{TP}{TP+FP}\times\frac{TP}{TP+FN}} \quad (12)$$

TP denotes true positive ratio signifying the number of positive cases which are correctly classified as positive. The positive case in this work refers to up signal. TN signifies true negative ratio that accounts for the number of negative cases (i.e. down signal) correctly classified as negative. On the other hand, FN denotes the number of positive cases misclassified as negative while FP implies the number of negative cases predicted as positives. Thus, magnitude of TP and TN should ideally be close to 1 for accurate classification whilst FP and FN values should be close to 0. Magnitudes of Specificity and Sensitivity should be close to 1 as well for models to be regarded as supreme.

G-Mean attempts to measure the balance between the performances of classifying positive and negative classes. Poor performance in correctly classifying positive cases would result in low G-mean value in spite of good accuracy in predicting negative cases. LP is positive likelihood ratio measuring the probability of classifying an instance as positive when it is negative actually and probability of classifying an actual positive instance as positive. LR reflects the opposite scenario, i.e. the ratio of probability of classifying an instance as negative when it is actually positive and probability of classifying a negative instance correctly. Higher LP and lower LR figures imply precise classification.

DP reflects the discriminant power of underlying classification models. DP values higher than 1 indicates supreme distinguishable capability. Youden's index (γ) and balanced accuracy (BA) figures should be close to 1 as well. Similarly, Mathews correlation coefficient (MCC), F1 score, and Fowlkes-Mallows (FM) index figures should lie close to 1 to infer high quality predictions. Apart from checking the classification accuracy, to measure the practical benefits of deploying FEB-Stacking and FEB-DNN models, trading benefits of both models have been estimated too.

4.7. Trading Benefits

To demonstrate the practical effectiveness of proposed framework a comparison with orthodox buy and hold (B&H) strategy has been conducted. The B&H strategy implies that the investor will invest a quantum of money in a particular stock and hold the same for a predefined time horizons, generally 3 months to 6 months duration. The net profit under this scheme is estimated after the completion of the time horizon. On the contrary, the proposed model suggests to invest for a predicted up (1) signal and to sell for predicted down (0) signal next day. The said process is continued for the entire time horizon. Thus the rate of return (ROR) can be calculated as:

$$ROR = \frac{net\ gain\ in\ stock}{initial\ investment} \tag{13}$$

Therefore, based on the estimated ROR figures, profitability of B & H strategy, FEB-Stacking, FEB-DNN respectively can be determined and relative performance can be measured. The said exercise has been performed for time horizon of 3 months at separate time horizons.

Finally to perform comparative statistical analysis with various other models, Diebold-Mariano's pairwise test for equal predictive ability has been evoked.

5. Results and Discussions

Executing classification exercise requires designing of training and test data segments systematically. Since we have two set of data samples Sets A and B, for critically evaluating the performance of FEB-Stacking and FEB-DNN on pre-Covid and post-Covid contexts respectively, training and test partitions have been formed for both sets in order to ascertain the predictive capabilities during normal and new-normal time horizons. The segmentation is made in forward looking direction which has been reported to be successfully utilized for time series prediction (Ghosh et al., 2019). For Set A observations ranging from January, 2014 to December, 2018 constitute training data points where as test segment comprises of observations from January, 2019 to December, 2019. The said partitioning evaluates the classification accuracy of FEB-Stacking and FEB-DNN models during the pre-Covid time horizons characterized by relatively low volatility and uncertainty. On the other hand, observations of January, 2014 to December, 2019 compose the training samples whilst data points spanning from January, 2020 to July, 2020 make up the test segment for Set B. The designed segmentation of Set B sample measures the predictive ability of respective models during the time period where the Covid-19 pandemic wreaked havoc.

As discussed, Stacking is implemented by combing output of RF, bagging, GB methods. These methods however are governed by several process parameters. To identify the most competent setting of hyper-parameters, the 'Gridserach' tool available at Keras interface has been evoked. All three constituent ensemble models are highly sensitive to parameters viz. number of base estimators, number of features

for branching operations of base learners, leaf nodes, etc. Using 'Gridsearch' utility these parameters can be varied and combinatorial search operation is performed to select the most prominent combination. On contrary, DNN with 2 hidden layers comprising 30 nodes each have been selected for learning process. Rectified Linear (Relu) activation function has been used at input and hidden layers whilst Linear activation function has been applied at output layer. Selection of batch size, number of iterations, and optimizer for learning process has been made through performing Gridsearch utility of Keras. The well-known 'Adam' optimizer has been found to be the optimal one.

5.1. Predictive Accuracy

The following figures 4-7 exhibit the resultant ROC plots alongside AUC values for FEB-Stacking and FEB-DNN models on Sets A and B.

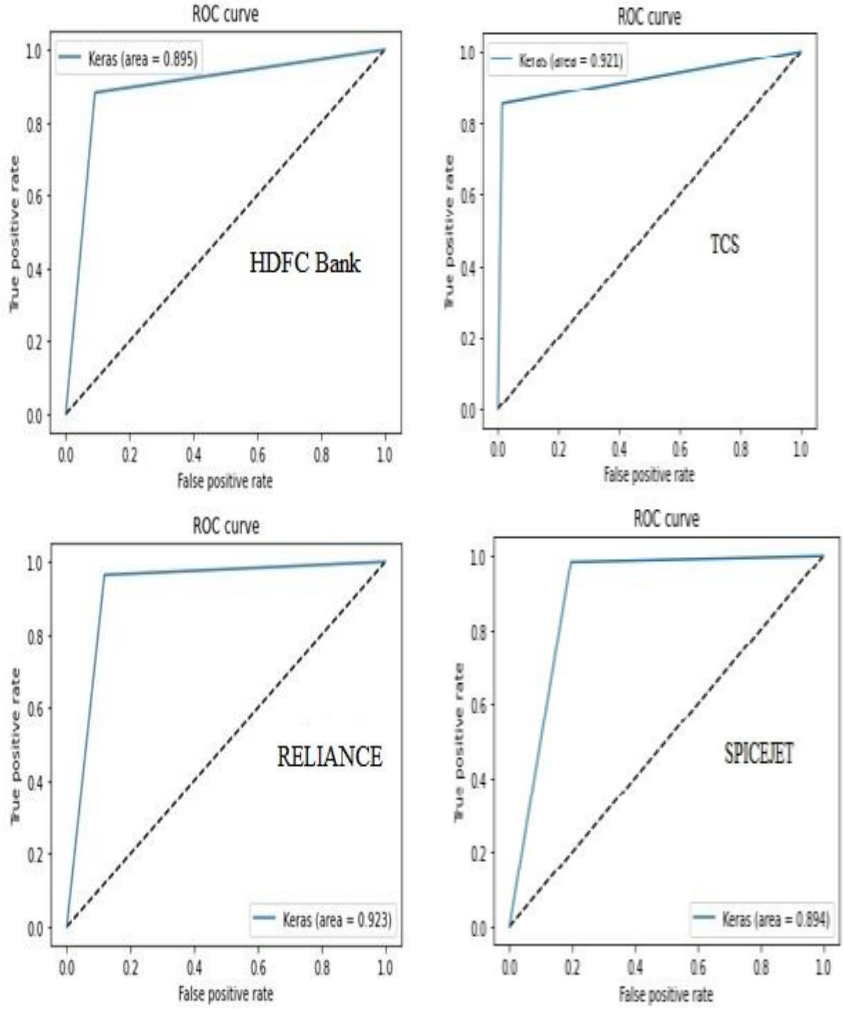

Figure 4. ROC Curve of FEB-Stacking on Test Segment of Set A Observations

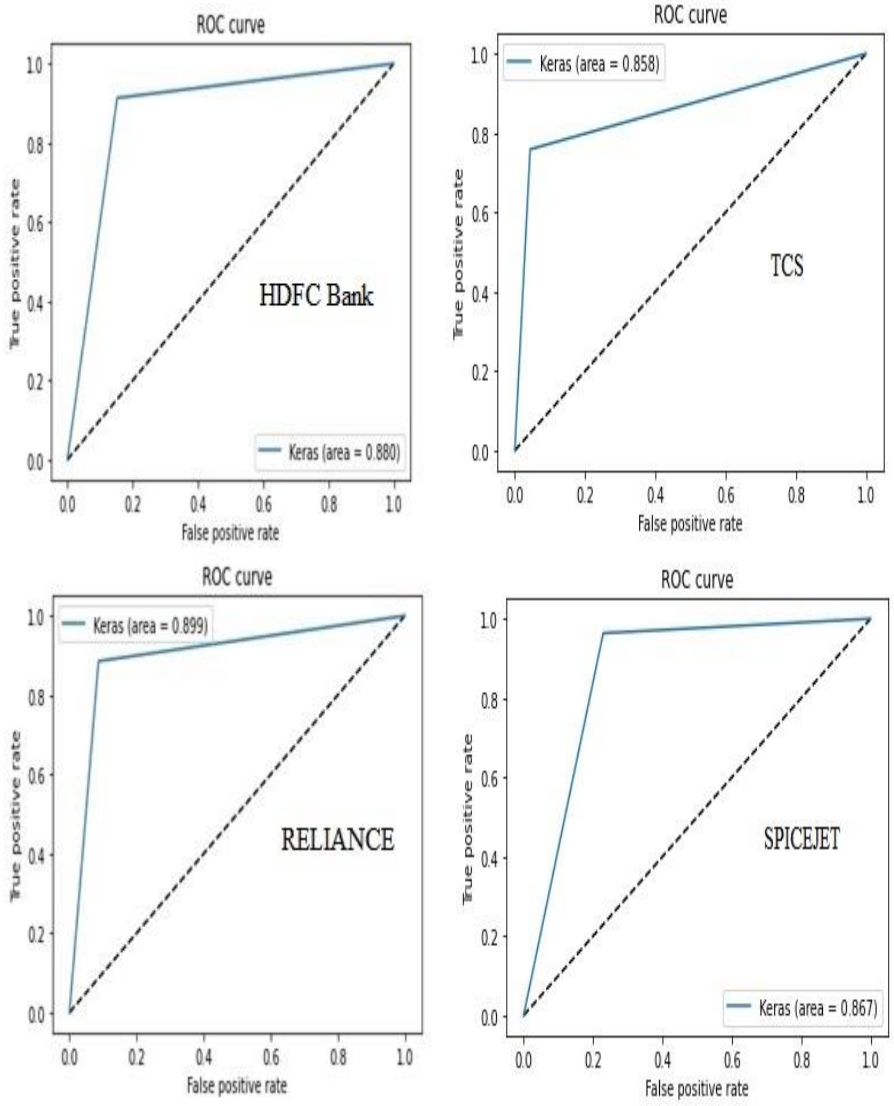

Figure 5. ROC Curve of FEB-Stacking on Test Segment of Set B Observations

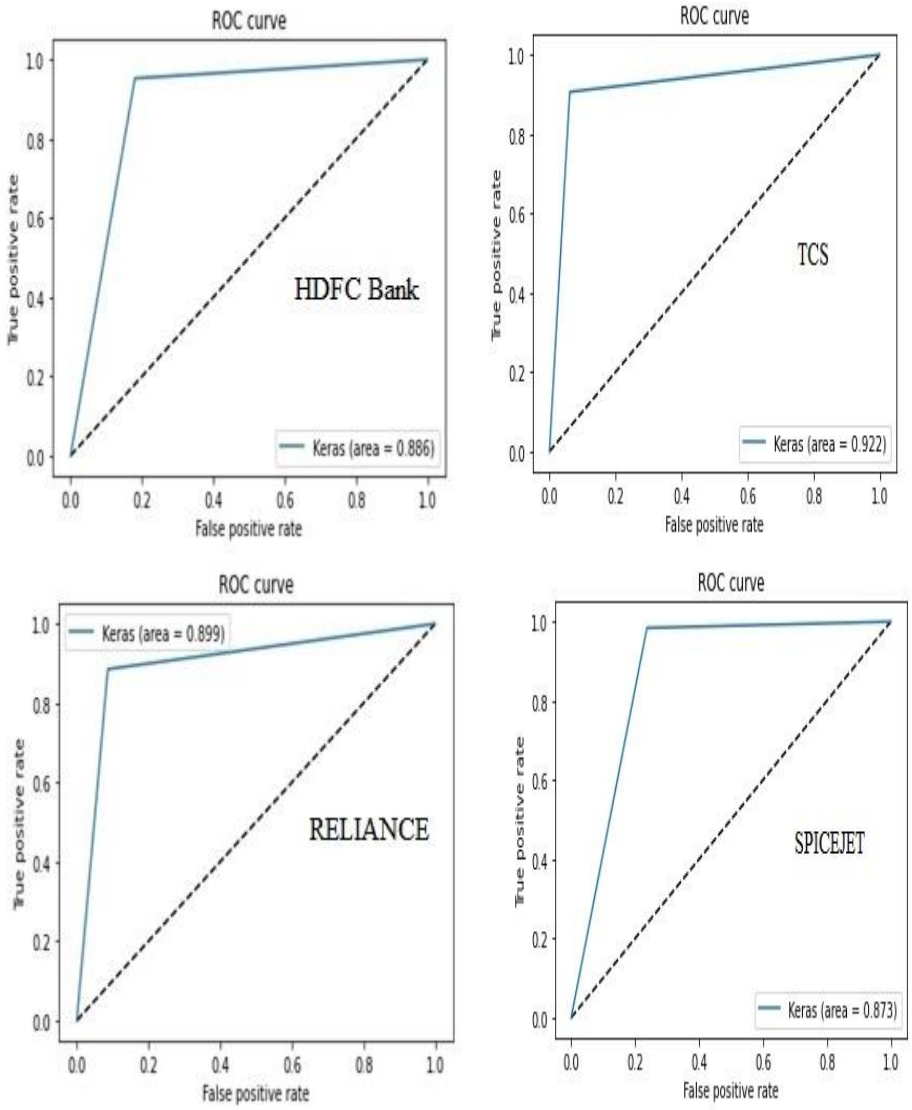

Figure 6. ROC Curve of FEB-DNN on Test Segment of Set A Observations

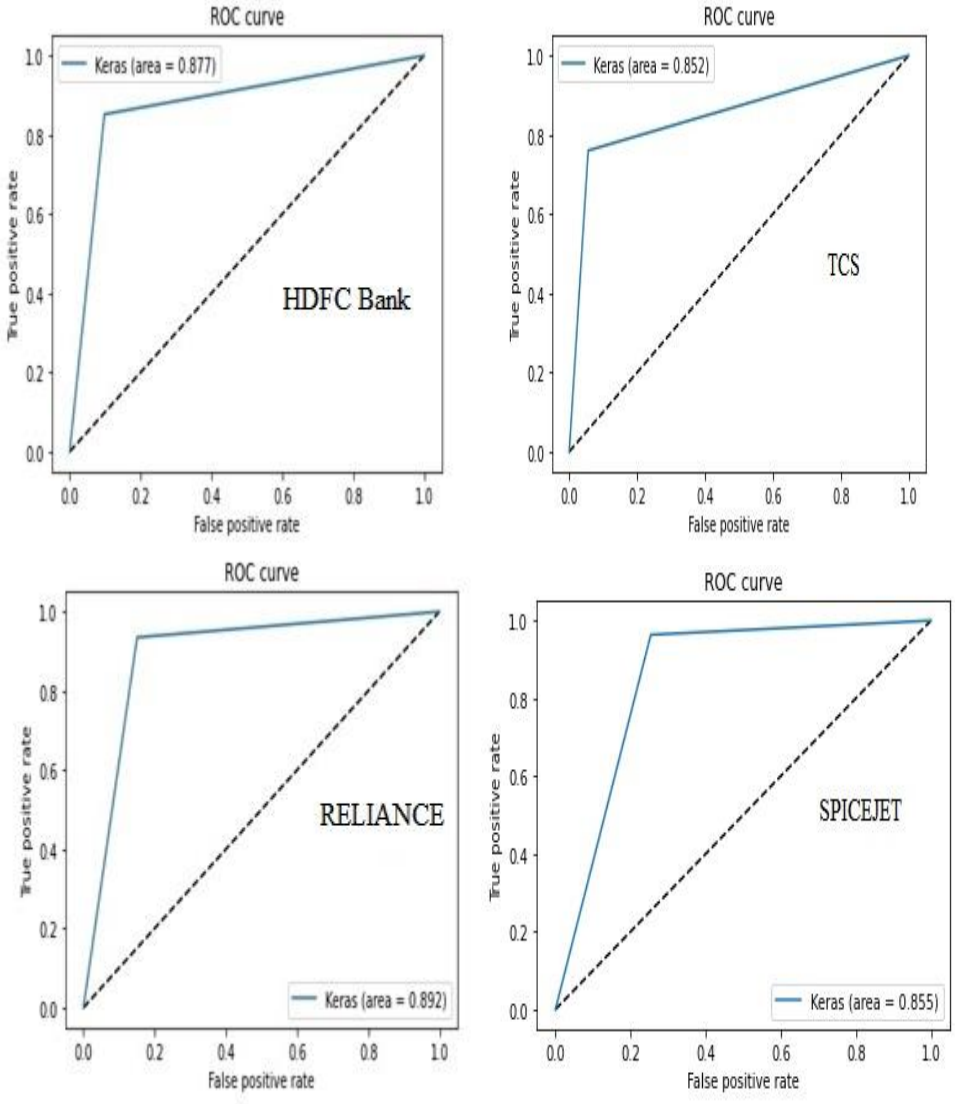

Figure 7. ROC Curve of FEB-DNN on Test Segment of Set B Observations

It can be noticed visually that AUC (represented by area in figures) values of resultant ROC curves on test data segments of pre-Covid and post-Covid periods have emerged to be pretty high for both FEB-Stacking and FEB-DNN models which basically implies good trend prediction performance. Nevertheless, to validate the inference drawn based on visual metrics, quantitative indices are estimated as well and presented in tables 5-8. At first, Table 5 outlines summary of performance of FEB-Stacking on Set A samples.

Table 5. Predictive Performance of FEB-Stacking on Set A

	HDFC Bank	TCS	RELIANCE	SPICEJET
		Training Data Set		
$Sensitivity$	0.9870	0.9931	0.9913	0.9870
$Specificity$	0.9121	0.9241	0.9226	0.9108
G	0.9488	0.9580	0.9563	0.9481
LP	11.2563	11.6528	11.5796	11.065
LR	0.0143	0.0075	0.0094	0.0143
DP	1.5962	1.7876	1.7266	1.5924
γ	0.8991	0.9172	0.9139	0.8978
BA	0.9495	0.9586	0.9569	0.9489
MCC	0.8102	0.8346	0.8328	0.8093
$F1$	0.8979	0.9186	0.9164	0.8972
FM	0.8935	0.9175	0.9170	0.8928
AUC	0.906	0.934	0.935	0.901
		Test Data Set		
$Sensitivity$	0.9759	0.9801	0.9790	0.9748
$Specificity$	0.9086	0.9154	0.9139	0.9086
G	0.9417	0.9472	0.9458	0.9411
LP	10.6795	10.7046	10.6794	10.665
LR	0.0264	0.0215	0.0219	0.0277
DP	1.4355	1.5027	1.4849	1.4246
γ	0.8846	0.8955	0.8929	0.8834
BA	0.9423	0.9477	0.9465	0.9417
MCC	0.7910	0.8247	0.8232	0.7903
$F1$	0.8823	0.9078	0.9066	0.8807
FM	0.8824	0.9096	0.9073	0.8811
AUC	0.895	0.921	0.923	0.894

It can be noticed that values of performance indicators on both training and test samples clearly lie on the zone which simply indicate remarkable performance of FEB-Stacking framework in carrying out directional predictive modeling of stock prices of HDFC Bank, TCS, RELIANCE, and SPICEJET during the pre-Covid time horizons. Values of sensitivity, specificity, $\boldsymbol{G}$, $\boldsymbol{\gamma}$, $\boldsymbol{BA}$, $\boldsymbol{MCC}$, $\boldsymbol{F1}$,$\boldsymbol{FM}$,and $\boldsymbol{AUC}$ have emerged to be close to 1. Superior capability of the proposed framework in distinctly predicting up and down trend can be inferred. High values of $\boldsymbol{LP}$ and low values of $\boldsymbol{LR}$ further solidify the claim. Therefore, it can be concluded that before the outbreak of Covid, i.e. in pre-Covid scenario, FEB-Stacking has accurately predicted future movements of HDFC Bank, TCS, RELIANCE, and SPICEJET stocks. We next, examine the performance of FEB-Stacking framework on trend prediction of underlying stocks on Set B dataset reflecting the scare part of Covid-19 pandemic. Table 6 summarizes the said findings.

Table 6. Predictive Performance of FEB-Stacking on Set B

	HDFC Bank	TCS	RELIANCE	SPICEJET
		Training Data Set		
$Sensitivity$	0.9778	0.9437	0.9789	0.9743
$Specificity$	0.9091	0.8785	0.9167	0.9069
G	0.9428	0.9105	0.9473	0.9400
LP	10.8248	10.5239	10.6918	10.4651
LR	0.0238	0.0254	0.0214	0.0283
DP	1.4571	1.1482	1.4924	1.4149
γ	0.8870	0.8221	0.9473	0.8812
BA	0.9435	0.9111	0.9478	0.9406
MCC	0.7926	0.7711	0.7975	0.7904
$F1$	0.8936	0.8692	0.9018	0.8917
FM	0.8841	0.8677	0.8924	0.8813
AUC	0.889	0.869	0.908	0.878
		Test Data Set		
$Sensitivity$	0.9683	0.9357	0.9779	0.9587
$Specificity$	0.8987	0.8698	0.9086	0.8914
G	0.9329	0.9021	0.9426	0.9244
LP	9.5587	7.1866	10.6991	8.8278
LR	0.0322	0.0325	0.0243	0.0463
DP	1.3408	1.0954	1.4568	1.2565
γ	0.8670	0.8055	0.8865	0.8501
BA	0.9335	0.9028	0.9433	0.9251
MCC	0.7819	0.7625	0.7810	0.7737
$F1$	0.8847	0.8611	0.8833	0.8788
FM	0.8768	0.8590	0.8857	0.8695
AUC	0.880	0.858	0.899	0.867

Like the performance on Set A, efficacy of FEB-Stacking framework in trend modeling is apparent on Set B as well as manifested by the figures of chosen performance indicators. However, it must be noted that the classification performance has marginally deteriorated as drop in sensitivity, specificity, $\boldsymbol{G}$, $\boldsymbol{LR}$, $\boldsymbol{\gamma}$, $\boldsymbol{BA}$, $\boldsymbol{MCC}$, $\boldsymbol{F1}$, and $\boldsymbol{FM}$ values can be observed whilst an increase in magnitude of $\boldsymbol{LR}$ is imminent on both training and test samples. The outcome is expected and logical due to the unprecedented shock induced by Covid-19 pandemic. Nevertheless, the figures of all these measures indeed indicate predictions of superior quality. Therefore, the framework can be regarded to be extremely efficient to yield predictions at extreme events as well. Subsequently, we evaluate the predictive capability of FEB-DNN on Set A and Set B datasets. Table 7 reports outcome of predictive performance FEB-DNN on Set A samples.

Table 7. Predictive Performance of FEB-DNN on Set A

	HDFC Bank	TCS	RELIANCE	SPICEJET
		Training Data Set		
$Sensitivity$	0.9896	0.9915	0.9904	0.9861
$Specificity$	0.9139	0.9227	0.9210	0.9095
G	0.9510	0.9565	0.9551	0.9470
LP	11.2601	11.6509	11.5781	10.8961
LR	0.0174	0.0149	0.0155	0.0153
DP	1.6557	1.7325	1.6975	1.5661
γ	0.9035	0.9142	0.9114	0.8956
BA	0.9518	0.9571	0.9557	0.9478
MCC	0.8132	0.8297	0.8328	0.8104
$F1$	0.9034	0.9159	0.9164	0.8988
FM	0.8976	0.9144	0.9170	0.8943
AUC	0.895	0.933	0.905	0.880
		Test Data Set		
$Sensitivity$	0.9783	0.9794	0.9753	0.9757
$Specificity$	0.9097	0.9138	0.9107	0.9081
G	0.9434	0.9460	0.9424	0.9413
LP	10.8615	10.6780	10.6772	10.6170
LR	0.02489	0.0233	0.0225	0.0268
DP	1.4644	1.4893	1.4356	1.4320
γ	0.8880	0.8932	0.8860	0.8838
BA	0.9440	0.9466	0.9430	0.9419
MCC	0.7966	0.8213	0.8209	0.7943
$F1$	0.8875	0.9044	0.9052	0.8837
FM	0.8849	0.9061	0.9059	0.8811
AUC	0.886	0.922	0.899	0.873

Similar to FEB-Stacking, predictive performance of FEB-DNN has emerged to be of supreme quality as manifested by the estimated classification indicators on both training and test data segments. Hence, FEB-DNN too can be regarded to be an extremely effective tool for trend prediction of chosen stocks during the normal time horizon i.e, pre-Covid time frame. Table 8 reports quality of performance on Set B samples.

Table 8. Predictive Performance of FEB-DNN on Set B

	HDFC Bank	TCS	RELIANCE	SPICEJET
		Training Data Set		
$Sensitivity$	0.9769	0.9439	0.9794	0.9548
$Specificity$	0.9086	0.8774	0.9186	0.8983
G	0.9428	0.9105	0.9473	0.9261
LP	10.8237	10.5221	10.6927	9.3884
LR	0.0254	0.0639	0.0224	0.0503
DP	1.4459	1.1467	1.5043	1.2515
γ	0.8870	0.8221	0.9473	0.8531
BA	0.9435	0.9111	0.9478	0.9402
MCC	0.7914	0.7706	0.7994	0.7876
$F1$	0.8922	0.8683	0.9031	0.8879
FM	0.8829	0.8668	0.8936	0.8792
AUC	0.884	0.861	0.899	0.871
		Test Data Set		
$Sensitivity$	0.9657	0.9312	0.9788	0.9489
$Specificity$	0.8964	0.8673	0.9101	0.8936
G	0.9329	0.9021	0.9426	0.9208
LP	9.3214	7.0173	10.8877	8.9182
LR	0.0383	0.0793	0.0233	0.0572
DP	1.3153	1.0729	1.4713	1.0965
γ	0.8670	0.8055	0.8865	0.8425
BA	0.9335	0.9028	0.9433	0.9213
MCC	0.7804	0.7598	0.7835	0.7746
$F1$	0.8829	0.8587	0.8856	0.8723
FM	0.8747	0.8573	0.8874	0.8683
AUC	0.877	0.852	0.892	0.855

Inspection of classification exercise on dataset carrying impact of Covid-19 pandemic reveals similar phenomenon observed in FEB-Stacking model. Classification performance of FEB-DNN model in Set B has seen a marginal drop in accuracy as compared to Set A. However, the overall figures of the indicators on both training and test samples does suggest that FEB-DNN has achieved noteworthy performance on highly volatile and uncertain time horizons affected by Covid-19 pandemic.

5.2. Profitability Analysis

To evaluate trading benefits of proposed schemes, FEB-Stacking and FEB-DNN, samples of approximately 1 month periods have been selected. During the selected time intervals B&H strategy is invoked to estimate the ROR%. Finally, ROR% based on predictions made by FEB-Stacking and FEB-DNN has been separately computed to perform a buy operation when trend of next day is predicted to be 'up' (1) and sell operation if predicted trend of next day is 'down' (0). The said exercises have been

repeated on three different time windows to evaluate the trading benefits of respective models. Table 9 reports the findings.

Table 9. Outcome of Profitability Analysis

	HDFC Bank	TCS	RELIANCE	SPICEJET
Time Period	1/9/2016 – 7/10/2016			
ROR% of FEB-Stacking	13.52%	18.16%	31.25%	11.26%
ROR% of FEB-DNN	13.41%	18.31%	31.09%	10.98%
ROR% of B&H Strategy	0.067%	-5.57%	7.75%	-3.46%
Time Period	1/7/2019-6/8/2019			
ROR% of FEB-Stacking	9.86%	11.20%	8.54%	8.17%
ROR% of FEB-DNN	9.77%	11.35%	8.69%	7.84%
ROR% of B&H Strategy	-11.55%	-1.10%	-11.05%	-12.61%
Time Period	6/5/2020-8/6/2020			
ROR% of FEB-Stacking	6.59%	11.19%	13.97%	5.32%
ROR% of FEB-DNN	6.46%	10.97%	13.91%	5.24%
ROR% of B&H Strategy	-29.71%	-12.05%	-9.78%	-24.66%

Time periods have been chosen randomly by critically covering the pre-Covid and post-Covid time horizons. First two samples assess the trading benefits of proposed models on normal time periods whilst the third sample evaluates profitability during new normal periods. Results clearly suggest dominance of both FEB-Stacking and FEB-DNN models over the orthodox B&H strategy as estimated ROR% figures of both models are substantially higher than the latter one on all three occasions. Outcome of profitability analysis is of paramount significance for investors as the proposed prediction models have emerged to yield substantial amount of profit even during the time of unprecedented circumstances owing to Covid-19 outbreak. Performance turned out to be exceptionally superior as compared to B&H strategy for normal time span as well. Among the stocks, RELIANCE has emerged to be most profitable which basically implies its superior performance in turbulent time as well. On the flipside, SPICEJET has turned out to be relatively less profitable, in comparison to the counterparts suggesting low confidence of investors. It must be noted that the proposed frameworks are tailor made for evaluation through ROR% to comprehend trading benefits. Inspection of risk related performance is beyond the scope of present work.

5.3. Comparative Performance Analysis

To ascertain the rationale of development of FEB-Stacking and FEB-DNN model, RF, ANN, multiple adaptive regression splines (MARS), support vector machine (SVM), and recurrent neural network (RNN) models have been applied to perform predictive modeling on Set B segment using same set of explanatory features as well. However, exclusive feature engineering through KPCA and bootstrapping operations are not attached to these models. 'Gridsearch' utility nevertheless has been utilized for finding optimal hyper-parameters of competing models. DM pairwise test has been evoked to perform pairwise comparison of underlying models. Since the test operates in a pairwise format and the outcome depends on the order of components, competing models are stacked with the index numbers for referring the order in the table for ease of comprehension. A significant positive test statistic figure signifies that the performance of second model is statistically superior to the first model. If test statistic value appears to be significantly negative then opposite scenario prevails, i.e. the superiority of the first model over the second model is implied. Tables 10-13 report the outcome of DM test.

Table 10. Comparative Performance Assessment on HDFC Bank

Models	RF (1)	ANN (1)	MARS (1)	SVM (1)	RNN (1)	FEB-Stacking (1)	FEB-DNN (1)
RF (2)	-						
ANN (2)	0.196#	-					
MARS (2)	0.203#	0.208#	-				
SVM (2)	0.191#	0.198#	0.221#	-			
RNN (2)	0.214#	0.213#	0.202#	0.228#	-		
FEB-Stacking (2)	6.9482***	6.9678***	6.9843***	6.9680***	6.9396***	-	
FEB-DNN (2)	6.9458***	6.9615***	6.9856***	6.9685***	6.9416***	0.195#	-

Not significant, *** Significant at 1% level of significance

Table 11. Comparative Performance Assessment on TCS

Models	RF (1)	ANN (1)	MARS (1)	SVM (1)	RNN (1)	FEB-Stacking (1)	FEB-DNN (1)
RF (2)	-						
ANN (2)	0.194#	-					
MARS (2)	0.215#	0.217#	-				
SVM (2)	0.198#	0.194#	0.229#	-			
RNN (2)	0.222#	0.226#	0.234#	0.210#	-		
FEB-Stacking (2)	6.9536***	6.9614***	6.9917***	6.9759***	6.9421***	-	
FEB-DNN (2)	6.9567***	6.9622***	6.9895***	6.9782***	6.9457***	0.192#	-

Not significant, *** Significant at 1% level of significance

Table 12. Comparative Performance Assessment on Reliance

Models	RF (1)	ANN (1)	MARS (1)	SVM (1)	RNN (1)	FEB-Stacking (1)	FEB-DNN (1)
RF (2)	-						
ANN (2)	0.213#	-					
MARS (2)	0.197#	0.220#	-				
SVM (2)	0.223#	0.203#	0.224#	-			
RNN (2)	0.235#	0.218#	0.211#	0.241#	-		
FEB-Stacking (2)	6.9488***	6.9646***	6.9724***	6.9693***	6.9408***	-	
FEB-DNN (2)	6.9484***	6.9679***	6.9708***	6.9687***	6.9443***	0.189#	-

Not significant, *** Significant at 1% level of significance

Table 13. Comparative Performance Assessment on SPICEJET

Models	RF (1)	ANN (1)	MARS (1)	SVM (1)	RNN (1)	FEB-Stacking (1)	FEB-DNN (1)
RF (2)	-						
ANN (2)	0.207#	-					
MARS (2)	0.193#	0.204#	-				
SVM (2)	0.211#	0.189#	0.229#	-			
RNN (2)	0.229#	0.213#	0.232#	0.226#	-		
FEB-Stacking (2)	6.9276***	6.9519***	6.9631***	6.9617***	6.9359***	-	
FEB-DNN (2)	6.9327***	6.9608***	6.9674***	6.9622***	6.938***	0.196#	-

Not significant, *** Significant at 1% level of significance

Sign and significance levels of DM test statistics clearly imply that FEB-Stacking and FEB-DNN have resulted in statistically superior trend predictions for all four underlying stocks, HDFC Bank, TCS, RELIANCE, and SPICEJET as compared to the five other models. On the other hand, no clear statistical evidence can be found to discriminate the performance of competing models. Therefore, outcome of comparative study clearly suggests supremacy of both FEB-Stacking and FEB-DNN over the remaining competing models in precisely estimating the trend of selected stocks in challenging times. Therefore, the importance of performing feature engineering bootstrapping apart from using high end stacking and DNN models is also justified. Hence, both FEB-Stacking and FEB-DNN frameworks have emerged to be extremely efficient and precise estimation of stock trends in normal and new-normal time horizons. Specifically, the performance during the Covid-19 pandemic is noteworthy and can immensely benefit traders and investors.

Our findings reveal that both frameworks, FEB-Stacking and FEB-DNN have emerged to be highly successful in trend classification of HDFC Bank, TCS, RELIANCE, and SPICEJET on both set of exercises. Quality of predictions during pre-Covid period has emerged to be marginally superior to the predictions obtained in post-Covid period. Nevertheless, the proposed architectures statistically outperformed several benchmark predictive tools during the said period. The models have appeared to be

highly profitable for trading purposes as well during the Covid-19 outbreak. The predictive structures have successfully accomplished the research objectives and can be regarded to be a contribution to the existing trend prediction literature. The strength of both frameworks lies in seamless integration of feature engineering, bootstrapping, and pattern mining process. Both frameworks have emerged to be highly successful in generating precise estimates of future across pre-Covid and post-Covid regimes. As expected, during post-Covid time horizons, prediction accuracy has marginally suffered. With availability of future data samples, both FEB-Stacking and FEB-DNN models can be tested for quality of accuracy over a prolonged period affected by Covid pandemic. Both models require identification of explanatory features beforehand.

Other advanced deep learning models, GRU, LSTM, CNN, GNN, etc. have been reported to be extremely successful in stock trend prediction as discussed in the literature. These models are famous for automatic extraction of features for predictive modeling. The present work, nevertheless, relied upon standard DNN model for predictive exercise. Since a substantial effort was put to form explanatory features in the form of technical and macroeconomic indicators and subsequent feature refining through KPCA, conventional DNN has turned out to be extremely effective in estimating trends with superior precision. However it would be interesting to explore the efficacy of our feature engineering process with aforesaid state-of-the-art deep learning models for stock trend prediction problems.

6. Conclusion

The present paper addresses a practical research problem of predicting trend of stock prices, particularly in volatile times resulting from the Covid-19 pandemic. The developed frameworks have been found to be efficient in estimating future movements of prices of three major Indian stocks belonging to three different industry verticals. FEB-Stacking and FEB-DNN frameworks have performed quite well in trend predictions in pre-Covid and post-Covid periods. Although the performance of the proposed architectures marginally deteriorated in post-Covid period, quality of predictions still emerged to be statistically superior to several benchmark ones. Apart from yielding high quality trend estimations, both frameworks have been found to be profitable as well as compared to orthodox B&H strategy, even at the time of exceedingly high uncertainty and fear in market owing to Covid-19 pandemic. The key contributions of the paper are listed below.

- Usage of technical indicators together with carefully chosen macroeconomic variables as proxies for market fear, market sentiment, sector outlook, and raw material availability.

- Transforming the raw independent features comprising technical and macroeconomic indicators through KPCA driven FE process to refine and augment the explanatory capabilities of feature set in predicting stock price trends during pre-Covid and post-Covid phases.

- Deployment of bootstrapping method for sorting class imbalance problem for strengthening the predictive frameworks. Statistically, the contribution of both these steps have been found to be of paramount significance as both FEB-Stacking and FEB-DNN have outperformed the competitive models.

- The performance of FEB-Stacking and FEB-DNN has emerged to be better during pre-Covid period i.e. normal time horizons than the post-Covid period reflecting new-normal time span. Nevertheless, the predictive accuracy of proposed models has been

found to be statistically more superior to RF, ANN, MARS, SVM, and RNN in both time periods.

- Increased profitability from use of both frameworks indicate that they can be effectively utilized for trading purposes.

The present paper has used stock prices of three companies in the binary trend prediction problem. State-of-the-art deep learning algorithms viz. LSTM, CNN, GAN, etc. can be explored and compared with presented frameworks on trend predictions of wider variety stocks belonging to different sectors. In future, explainable AI can be added on top of predictive architectures to interpret the positive or negative influence of the explanatory features. Similarly the frameworks can easily be extended for trend modeling of different financial assets viz. foreign exchange and commodities. Additional class levels may be added to test the efficacy of proposed schemes in multiclass prediction problems.

Author Contributions: Each author has participated and contributed sufficiently to take public responsibility for appropriate portions of the content.

Funding: This research received no external funding.

Conflicts of Interest: The authors declare no conflicts of interest.

References

Ahmar, A. S. & Val, E. B. D. (2020). SutteARIMA: Short-term forecasting method, a case: Covid-19 and stock market in Spain. Science of The Total Environment, 729, 138883.

Atsalakis, G. S. & Valavanis, K. P. (2009). Forecasting stock market short-term trends using a neuro-fuzzy based methodology. Expert Systems With Applications, 36, 10696-10707.

Babu, C. N. & Reddy, B. E. (2015). Prediction of selected Indian stock using a partitioning–interpolation based ARIMA–GARCH model. Applied Computing and Informatics, 11, 130-143.

Basu, S. (1977). Investment Performance of Common Stocks in Relation to Their Price Earnings Ratios: A Test of the Efficient Market Hypothesis. Journal of Finance, 32, 663-682.

Basu, S. (1983). The Relationship between Earnings Yield, Market Value and Return for NYSE Common Stocks: Further Evidence. Journal of Financial Economics, 12, 129-156.

Bisoi, R., Dash, P. K. & Parida, A. K. (2019). Hybrid Variational Mode Decomposition and evolutionary robust kernel extreme learning machine for stock price and movement prediction on daily basis. Applied Soft Computing, 74, 652-678.

Breiman, L. (2001). Random forests. Machine Learning, 45, 5–32.

Bria, A., Marrocco, C. & Tortorella, F. (2020). Addressing class imbalance in deep learning for small lesion detection on medical images. Computers in Biology and Medicine, 120, 103735.

Carta, S., Ferreira, A., Podda, A. S., Recupero, D. R. & Sanna, A. (2021). Multi-DQN: An ensemble of Deep Q-learning agents for stock market forecasting. Expert Systems with Applications, 164, 113820.

Chatzis, S. P., Siakoulis, V., Petropoulos, A., Stavroulakis, E. & Vlachogiannakis, N. (2018). Forecasting stock market crisis events using deep and statistical machine learning techniques. Expert Systems With Applications, 112, 353-371.

Chen, A. S., Leung, M. T. & Daouk, H. (2003). Application of Neural Networks to an Emerging Financial Market: Forecasting and Trading the Taiwan Stock Index. Operations Research in Emerging Economics, 30, 901– 923.

Chen, Y. & Hao, Y. (2018). Integrating principle component analysis and weighted support vector machine for stock trading signals prediction. Neurocomputing, 321, 381-402.

Das, S. R., Mishra, D. & Rout, M. (2019). Stock market prediction using Firefly algorithm with evolutionary framework optimized feature reduction for OSELM method. Expert Systems with Applications: X, 4, 100016.

Datta Chaudhuri, T., Ghosh, I. & Eram, S. (2016). Application of Unsupervised Feature Selection, Machine Learning and Evolutionary Algorithm in Predicting Stock Returns: A Study of Indian Firms. IUP Journal of Financial Risk Management, 13, 20-46.

Datta Chaudhuri, T., Ghosh, I. & Singh, P. (2017). Application of Machine Learning Tools in Predictive Modeling of Pairs Trade in Indian Stock Market. IUP Journal of Applied Finance, 23, 5-25.

Dutta, G., Jha, P., Laha, A. & Mohan, N. (2006). Artificial Neural Network Models for Forecasting Stock Price Index in the Bombay Stock Exchange. Journal of Emerging Market Finance, 5, 283-295.

Fama, E. F. & French, K. R. (1988). Dividend Yields and Expected Stock Returns. Journal of Financial Economics, 1, 3-25.

Fama, E. F. & French, K. R. (1992). The Cross-Section of Expected Stock Returns. Journal of Finance, 47, 427-465.

Fama, E. F. & French, K. R. (1995). Size and Book-to-Market Factors in Earnings and Returns. Journal of Finance, 50, 131-155.

Ghosh, I. & Datta Chaudhuri, T. (2018). Stock Market Portfolio Construction: A Four-stage Model Based on Fractal Analysis. South Asian Journal of Management, 25, 117-149.

Ghosh, I., Jana, R. K., & Sanyal, M. K. (2019). Analysis of temporal pattern, causal interaction and predictive modeling of financial markets using nonlinear dynamics, econometric models and machine learning algorithms. Applied Soft Computing, 82, 105553.

Ghosh, I., Sanyal, M. K. & Jana, R. K. (2018). Fractal inspection and machine learning-based predictive modelling framework for financial markets. Arabian Journal for Science and Engineering, 43, 4273-4287.

Graham, B. & Dodd, D. (1934). Security Analysis. 1st Edition, Mcgraw Hill, New York.

Ibbotson, R. & Idzorek, T. (2014). Dimensions of Popularity. Journal of Portfolio Management, 40, 68-74.

Ismail, M. S., Noorani, M. S. M., Ismail, M., Razak, F. A. & Alias, M. A. (2020). Predicting next day direction of stock price movement using machine learning methods with persistent homology: Evidence from Kuala Lumpur Stock Exchange. Applied Soft Computing, 93, 106422.

Jaffe, J., Keim, D. B. & Westerfield, R. (1989). Earnings Yields, Market Values, and Stock Returns. Journal of Finance, 44, 135-148.

Jiang, W. (2020). Applications of deep learning in stock market prediction: recent progress. arXiv preprint arXiv:2003.01859.

Kingma, D. P. & Ba, J. (2017). Adam: A Method for Stochastic Optimization. arXiv:1412.6980.

Lariviere, B. & Van den Poel, D. (2005). Predicting customer retention and profitability by random forests and regression forests techniques. Expert Systems with Applications, 29, 472–484.

Lei, L. (2018). Wavelet Neural Network Prediction Method of Stock Price Trend Based on Rough Set Attribute Reduction. Applied Soft Computing, 62, 923-932.

Lemmens, A. & Croux, C. (2006). Bagging and Boosting Classification Trees to Predict Churn. Journal of Marketing Research, 43, 276-286.

Liu, H. & Long, Z. (2020). An improved deep learning model for predicting stock market price time series. Digital Signal Processing, 102, 102741.

Liu, M., Wang, M., Wang, J. & Li, D. (2013). Comparison of random forest, support vector machine and back propagation neural network for electronic tongue data classification: Application to the recognition of orange beverage and Chinese vinegar. Sensors and Actuators B: Chemical, 177, 970–980.

Liu, W., Wang, Z., Liu, X., Zeng, N., Liu, Y. & Alsaadi, F. E. (2017). A survey of deep neural network architectures and their applications. Neurocomputing, 234, 11-26.

Mostafa, M. (2010). Forecasting Stock Exchange Movements using Neural Networks: Empirical evidence from Kuwait. Expert Systems with Application, 37, 6302-6309.

Perez-Rodriguez, J. V., Torra, S. & Andrada-Felix, J. (2005). Star and ANN models: Forecasting performance on the Spanish Ibex-35 stock index. Journal of Empirical Finance, 12, 490–509.

Pirizadeh, M., Alemohammad, N., Manthouri, M. & Pirizadeh, M. (2020). A new machine learning ensemble model for class imbalance problem of screening enhanced oil recovery methods. Journal of Petroleum Science and Engineering, https://doi.org/10.1016/j.petrol.2020.108214.

Qureshi, A. S., Khan, A., Zameer, A. & Usman, A. (2017). Wind Power Prediction using Deep Neural Network based Meta Regression and Transfer Learning. Applied Soft Computing, 58, 742-755.

Rundo, F., Trenta, F., di Stallo, A. L. & Battiato, S. (2019). Machine learning for quantitative finance applications: A survey. Applied Sciences, 9, 5574.

Schapire, R. E. and Singer, Y. (1999). Improved boosting algorithms using confidence-rated predictions. Machine Learning, 37, 297-336.

Scholkopf, B., Smola, A. & Muller, K. R. (1999). Kernel principal component analysis. Advances in Kernel Methods – Support Vector Learning. MIT Press, Massachusetts. (Chapter 2)

Sezer, O, B., Gudelek, M. U. & Ozbayoglu, A. M. (2020). Financial time series forecasting with deep learning: A systematic literature review: 2005–2019. Applied Soft Computing, 90, 106181.

Shen, J., Fan, H. & Chang, S. (2007). Stock Index Prediction Based on Adaptive Training and Pruning Algorithm. Advances in Neural Networks, 4492, 457–464.

Shin, J., Yoon, S., Kim, Y. W., Kim, T., Go, B. G. & Cha, Y. K. (2021). Effects of class imbalance on resampling and ensemble learning for improved prediction of cyanobacteria blooms. Ecological Informatics, 61, 101202.

Simidjievski, N., Todorovski, L. & Dzeroski S (2015). Predicting long-term population dynamics with bagging and boosting of process-based models. Expert Systems with Applications, 42, 8484-8496.

Srinivasan, P. & Prakasam, K. (2014). Gold Price, Stock Price and Exchange Rate Nexus: The Case of India. IUP Journal of Financial Risk Management, 11, 52-62.

Strong, N. & Xu, X. G. (1997). Explaining the Cross-Section of UK Expected Stock Returns. The British Accounting Review, 29, 1-23.

Zhang, X. D., Li, A. & Pan, R. (2016). Stock trend prediction based on a new status box method and AdaBoost probabilistic support vector machine. Applied Soft Computing, 49, 385-398.

Zheng, Y., Caixin, S., Jian, L., Qing, Y. & Weigen, C. (2011). Entropy-Based Bagging for Fault Prediction of Transformers Using Oil-Dissolved Gas Data. Energies, 4, 1138-1147.

Zhou, F., Zhou, H. M., Yang, Z. & Yang, L. (2019). EMD2FNN: A strategy combining empirical mode decomposition and factorization machine based neural network for stock market trend prediction. Expert Systems with Applications, 115, 136-151.