MDPI

Article

Can Ensemble Machine Learning Methods Predict Stock Returns for Indian Banks Using Technical Indicators?

Sabyasachi Mohapatra [1,*], Rohan Mukherjee [1], Arindam Roy [2], Anirban Sengupta [1] and Amit Puniyani [3]

1   Indian Institute of Management, Bodhgaya 824234, India
2   Birla Institute of Technology & Science, Pilani 333031, India
3   Goa Institute of Management, Goa 403505, India
*   Correspondence: sabya@iimbg.ac.in

Abstract: This paper develops ensemble machine learning models (XGBoost, Gradient Boosting, and AdaBoost in addition to Random Forest) for predicting stock returns of Indian banks using technical indicators. These indicators are based on three broad categories of technical analysis: Price, Volume, and Turnover. Various error metrics like Mean Absolute Error (MAE), Mean Squared Error (MSE), Mean Absolute Percentage Error (MAPE), Root-Mean-Squared-Error (RMSE) have been used to check the performance of the models. Results show that the XGBoost algorithm performs best among the four ensemble models. The mean of absolute error and the root-mean-square -error vary around 3–5%. The feature importance plots generated by the models depict the importance of the variables in predicting the output. The proposed machine learning models help traders, investors, as well as portfolio managers, better predict the stock market trends and, in turn, the returns, particularly in banking stocks minimizing their sole dependency on macroeconomic factors. The techniques further assist the market participants in pre-empting any price-volume action across stocks irrespective of their size, liquidity, or past turnover. Finally, the techniques are incredibly robust and display a strong capability in predicting trend forecasts, particularly with any large deviations.

Keywords: ensemble techniques; machine learning; stock prediction; Indian banks

1. Introduction

Exploring the main drivers for stock market prediction has been one of the most sought-after areas in financial economics in recent decades. Traditional asset pricing models proposed by Fama and French (1993, 2015) attributed that stock returns are linked to underlying fundamental factors like market cap, Book-to-Market ratio, and systemic risk in a linear manner. Zhu et al. (2011) have proposed the workhorse for financial modeling. This model is used extensively by academicians as well as industry practitioners. However, uncertainty in the markets often challenges the assumption of linearity in the financial markets (as was assumed in the earlier models).

Realizing the humongous data involved corresponding to the various technical and fundamental variables, experts have hugely turned to the recent developments in machine learning methods in establishing an alternative paradigm for studying the relationship between company-relevant features and the stock price forward process. This type of model is useful in providing more diverse approaches than traditional models (Neely et al. 2014; Dai et al. 2020). The literature has shown that by applying powerful modeling approaches, such as ANN (artificial neural networks) (Khashei and Bijari 2010; Alberg and Lipton 2017; Belciug and Sandita 2017), decision trees (Sorensen et al. 2000; Andriyashin et al. 2008; Zhu et al. 2011, 2012), deep neural networks (Chong et al. 2017; Kraus and Feuerriegel 2017), random forests (Krauss et al. 2017), the classification and prediction efficiency of stocks have increased significantly.

While most of the earlier asset pricing studies based on the risk premium theories have strongly emphasized the role of the macro and firm-specific indicators, little effort

has been put into evaluating the role of technical indicators (Neely et al. 2014; Wu et al. 2021; Cheng et al. 2022). Most of the macro indicators (like the change in the interest rates, gross domestic product, the balance of trade, and the consumer price index) used in the classical asset-pricing models are lagging indicators that were determined post the events. We understand that any change in these lagging indicators is witnessed only when the economy starts following a particular pattern. Even though these indicators might be more precise than the leading ones, any visible change is observed only after a large economic shift. As a result, classical asset pricing models developed on the macroeconomic variables are of limited usage, particularly to short-term investors and intra-day traders. The aforesaid understanding lays the foundation for evaluating the role of the technical indicators in predicting the stock returns and in turn the equity risk premium.

The evaluation of the technical indicators gains further prominence as the related variables change daily, indicating in particular, the mood of the short-term investors and intra-day traders. As a result, realizing the huge volume of data corresponding to the technical indicators generated on a daily frequency basis, we attempt to develop ensemble machine learning models for a precise prediction of the equity returns.

In the present study, we have explored ensemble machine learning models, namely (XGBoost, Gradient Boosting, and AdaBoost in addition to Random Forest) for predicting stock returns of Indian banks using technical indicators. The present work has explored the ensemble machine learning models, namely (XGBoost, Gradient Boosting, and AdaBoost in addition to Random Forest) for predicting stock returns of Indian banks using technical indicators. The literature has established that machine learning models outperform statistical and econometric models (Hsu et al. 2016; Meesad and Rasel 2013; Patel et al. 2015). One big advantage of machine learning techniques is that there is no need to justify distributional assumptions and the ability to recognize the hidden patterns of time-series data. Also, the reduction of variance in machine learning models and the gain in prediction accuracy have contributed to the popularity of those models in stock prediction. The work described in (Hsu et al. 2016) has vividly established the superiority of machine-learning techniques over non-machine-learning techniques for intraday and daily predictions across major markets. It has been further well-established in the literature that major drawbacks associated with the traditional statistical and econometric models have strongly encouraged the growth of machine learning and deep learning techniques for stock prediction (Zhang et al. 2015; Zhao et al. 2017; Chong et al. 2017; Kazem Ahad et al. 2013; Kristjanpoller and Michell 2018; Lei 2018). Further exploring the machine learning domain, the literature suggests that ensemble models can perform better than a single model in financial prediction systems (Ampomah et al. 2020; Nti et al. 2020). In such a context, the current research explores the ensemble techniques in prediction stocks in the Indian scenario using a particular sector as a reference.

As reported by the World Bank, India's GDP in dollar terms had grown all the way to USD 2.9 trillion in 2019 before plummeting to USD 2.7 trillion in 2020 owing to the COVID-19 impact. Despite the degrowth, the Indian Economy continued to retain its tag of the fastest growing major economy in the world. Currently, the Indian Republic is the largest democracy and the sixth largest economy in the world in terms of nominal GDP. At the current prices, India's nominal GDP is estimated to be at USD 3.12 trillion in FY22. The Indian economy is driven by an ideal mix of advanced industries and modern agriculture, showcasing the recent deliberations made by the Government (like reducing the minimum capital barrier in key sectors and simplifying the licensing process) to attract the FDI inflows further and facilitate the economic growth.

During this time, under the supervision of the capital market regulator, Securities and Exchange Bureau of India, the financial markets in India became more transparent and robust. This has brought the attention of global investors to the Indian capital markets, which is evident by the size of the National Stock Exchange (NSE). As of March 2022, NSE is ranked as the 9th largest stock exchange in terms of market cap (USD 3.45 trillion). In the due course of time, the weightage assigned to Indian equities in the widely followed

MSCI-emerging market index has also increased with time indicating an interest of global investors in the rising Indian economy. Such a tremendous growth of the Indian financial market particularly in the last decade has attracted the attention of the market participants, as well as academicians in unraveling the prediction of Indian stocks by applying machine learning techniques (Dutta et al. 2006; Kumar and Thenmozhi 2007; Panda and Narasimhan 2007; Patel et al. 2015).

To give a further distinction to our current study, we limit our evaluation to Indian Banks. There are three major reasons behind this approach. Firstly, banking as a sector plays a critical role in the development of emerging economies like India. The banking sector in India contributes 7–8% toward GDP. They account for most of the funding in the ongoing and future businesses particularly represented by the small and medium enterprises. Banks directly contribute to job creation and the overall development of the economy. Secondly, banks and financial services contribute almost 36 percentage points to the S&P Nifty 500 benchmark index. Big and large banking stocks from the public as well as the private sector majorly decide the movement and the overall returns of the benchmark index. Finally, The Indian banking system is all set to draw benefit out of the improved credit offtake post-COVID-19 and the resilient risk management practices lasting the trying times. Public as well as private sector banks continue to grow their widely spread branch network led by huge technology-led investments targeting a better customer outreach with an increased service level satisfaction. Both public as well as private sector banks target to increase their core profitability on the back of increasing operational efficiencies. Further, the finance industry has systematically looked for ways to predict future asset returns, based on financial time-series data. The focus is mainly to predict the sign, either for shorter or longer horizons rather than predicting the effective returns. However, this task has become challenging as markets are noisy and have volatile environments with fluctuations and shifts in volatilities. Realizing the challenges, in the current study, we uniquely develop ensemble machine learning models for predicting the stock returns solely on the evaluation of the Technical Indicators built using the Price, Volume, and Turnover features for the Indian Banks. The study is in line with (Chen et al. 2022), who used machine learning techniques for predicting the stock prices of Chinese banks.

From a methodology perspective, the uniqueness comes from the fact that the ensemble techniques have not been exploited in stock prediction in Indian financial markets. To the best of our knowledge, this is the first study that is applying ensemble methods, to be more specific the boosting algorithms on Indian stocks. The following sections mention the existing literature and the methodology design, followed by the results and conclusion.

2. Literature Review

Predicting the market over a relatively short period (on the scale of days) has traditionally involved the analysis of price and volume data using methods ranging from linear prediction models (which tests the efficient market hypothesis to more complex machine learning) and hybrid machine learning methods (Kim and Han (2000); Choudhry and Garg (2008); Guiso et al. (2008, 2011); Majhi et al. (2009); Kara et al. (2011); Zhu et al. (2011, 2012); Shen et al. (2012); Edmans et al. (2012); Ajmi et al. (2014); De Oliveira et al. (2013); Patel et al. (2015); Chou and Nguyen (2018); Kim et al. (2020)). The theory of efficient market hypothesis (Fama 1970) has suggested that markets efficiently price in all available information leaving the stock movements vulnerable only to events that cannot be predicted. This phenomenon is also observed by Hellström and Holmström (1998). White (1988) conducted a test of the efficient market hypothesis using an Artificial Neural Network (ANN). This is the first stock market study that used ANNs. Artificial Neural Networks are a dense network of neurons that are interconnected with each other. These interconnected neurons get activated based on the inputs. A similar study on six stocks listed on the NYSE using historical price data has been reported (Tsibouris and Zeidenberg 1995). Several other studies such as Kolarik and Rudorfer (1994) have noted that ANNs have performed better than statistical techniques such as ARIMA and regression. Guresen

et al. (2011) have studied a modified Multi-Layer Perceptron applied to the Turkish stock market and concluded that the classical ANN is more suitable for stock market prediction. Qiu and Song (2016), Qiu et al. (2016), Zhong and Enke (2017), Coyne et al. (2018), Hu et al. (2018) trained their respective models by applying the backpropagation technique for the MLP models.

ANN is one of the most applied techniques for predicting stock markets in recent times as well Bustos and Pomares-Quimbaya (2020). In recent works, Qiu and Song (2016), Qiu et al. (2016), Hu et al. (2018), there has been the dominant application of Artificial Neuron Networks for stock market prediction. Kara et al. (2011) have again used Support Vector Machine (SVM) and ANN to predict daily movement direction in the ISE (Istanbul stock exchange) and reported that ANNs are more powerful than SVMs. Chang (2011) has used ANNs and decision trees for digital game content stock price prediction in the Chinese stock market and obtained accuracy in the range of 13–15%. Fagner A. De Oliveira et al. (2013) applied ANNs to predict the highly liquid Brazilian oil stock and obtained reasonable accuracy. More lately, the works of Song et al. (2018) and Huang and Liu (2019) have used various hybrid and classical machine learning (ML) approaches to predict stock prices in the Chinese and Taiwanese stock markets.

Several authors have applied ML methods to the Indian stock market. Kumar and Thenmozhi (2007) used SVMs to forecast the CNX Nifty index return and reported greater accuracy of SVMs than with other methods. Panda and Narasimhan (2007) found that ANNs outperform various linear autoregressive models in predicting daily BSE Sensex prices. Dutta et al. (2006) computed the accuracy of ANNs in predicting weekly closing prices of the Mumbai stock exchange over a two-year period using RMSE and MAE measures which they found to be in the range of 4–6%. Very recent work on Indian stock data considering stocks of Infosys (IT sector), ICICI (banking sector), and Sun Pharmaceuticals (healthcare sector) compared time series (Holt–Winters Exponential Smoothing), one econometric model (ARIMA) and two machine Learning models (Random Forest and MARS). The study found the machine learning models to be superior (Chatterjee et al. 2021).

Recent work by Ciner (2019) employed an ensemble of decision tree models (Random Forest Method and Boosted Tree approach) to significantly moderate or completely eradicate any uninformative forecasters amongst extremely correlated variables to improve the prediction as compared to the traditional Ordinary Least Square (OLS) method. Park et al. (2022) proposed a prediction framework (LSTM-Forest) to address the overfitting issue confronted in the case of the Deep Learning models with an increase in the count of the relevant input variables. The multitask model proposed improved the predictability, returns, and profitability for S&P500, SSE Composite Index, and KOSPI200 as compared to the baseline Random Forest model. Andriyashin et al. (2008) applied the decision trees technique to a ternary classification of stocks, who are the constituents of DAX. They have trained the model on both the technical as well as fundamental variables. Similar tree-based models are employed in the works of Sorensen et al. (2000), Zhu et al. (2011, 2012). They have examined the stock classification efficiency. They are based on different feature spaces. The decision tree framework is superior to that of the linear weighting approach for explaining higher-order relationships between stock returns and underlying variables. Similarly, the decision tree framework is better than that of the linear weighting approach in providing risk-diversified portfolios. Krauss et al. (2017) has applied state-of-the-art machine learning techniques. They included DNN, GBDT, and Random forest as well as a combination of the three techniques (which are also known as ensemble techniques) to constituent stocks of the S&P 500. Similar works (by applying ensemble techniques) have been done recently (Nti et al. 2020; Ayala et al. 2021). Finally, Patel et al. (2015) used various ML techniques to predict the deterministic trend for the NIFTY index and various large caps and find that random forests outperform ANNs, SVMs, and naive Bayes models. Also, Naik and Mohan (2020) applied the deep learning method for forecasting stocks in the Indian financial market.

In recent years, Wang et al. (2012), Rasekhschaffe and Jones (2019), Leung et al. (2021), Hanauer et al. (2022) applied boosting techniques for stock price prediction. However, these studies are based on fundamental and macroeconomic variables. They have covered the Chinese and European Stock markets. In addition to country-specific models, sector-specific stock prediction models have emerged (Sadorsky 2021; Day et al. 2022; Challa et al. 2020). The sector-specific models help the investors and traders to pursue their stock selection strategy by applying the machine learning techniques proposed in various studies. This will help them earn higher alpha and beat the benchmark index by investing in the sector-specific index. One more benefit of having sector-specific investment models (based on machine learning) is that each sector has different fundamental characteristics and technical characteristics (price, price volatility, volume). Recently Chen et al. (2022) applied machine learning techniques for predicting the stock prices of Chinese banks. Prior to Chen et al. (2022), machine learning techniques and network models have gained significant importance in bankruptcy prediction (Shetty et al. 2022), exchange rate prediction (Zhang and Hamori 2020), bank corporate governance (Carlini et al. 2020), and assessing bank competition (Rahman and Misra 2021). However, our study is the first study to be conducted on the Indian stock market with several ensemble techniques, specifically the boosting algorithms.

3. Data Description

The paper contains data on the stock prices of publicly traded banks in India. The data ranges from 1 January 2014 to 31 December 2021. This period has been associated with many financial challenges as well as changes in the regulatory framework for the banks.

Between 2014–2021, the Indian banking system was badly hit by the rising NPA issues. Almost all the banks (especially the public sector banks) witnessed an unprecedented level of Gross NPAs. The gross NPAs of public sector banks almost doubled to the tune of USD 700 Million in 2021 from USD 325 Million. The central bank, that is, the Reserve Bank of India passed a regulation for recognition of the NPAs (non-performing assets) and cleaning of the balance sheets. Also, during this period, the Indian government not only administered demonetization to curb corruption and the related black money forming a part of the economy, but it also implemented major bills and laws like the Insolvency and Bankruptcy Code and the Goods and Service Tax affecting the NPAs as well as profitability of the banks. The study period also takes into account the financial disruption caused by the COVID-19 pandemic and its impact on the Indian economy and particularly the banking sector at large.

In this dataset, we have 10 input variables and one variable in the output space. These 10 variables are based on three broad variable classification in technical analysis: Price, Volume, Turnover (Neely et al. 2014; Dai et al. 2020). These variables are essentially from technical analysis feature space. The novelty of the input variables is that they broadly cover three important dimensions of technical analysis, which are trading volumes, stock turnover, and stock prices. The input variables are discussed below in Table 1.

In the output feature space, we have calculated the excess return of individual banking stock over NIFTY 500 return (represented by Equation (3)). Equation (1) measures the price return of particular stock for a 20-day holding period. Equation (2) measures the price return of the benchmark index (NIFTY 500) for a holding period of 20 days. Equation (3) measures the excess return of individual banking stock over the benchmark index (NIFTY 500) for a holding period of 20 days.

Table 1. Technical Factors—Description and Mathematical Computation.

Factors		Description	Mathematical Computation
10by20 10by40	- -	Parameter indicates growth in Trading Volume during the past '*t*' trading days; where *t* = 20 and 40. Help identify market participants' trading interests based on volume trendlines.	$\dfrac{\text{movavg (volume},m_1)}{\text{movavg (volume},m_2)}$ where, $m_1 \in (10)$, $m_2 \in (20, 40)$
MAT20 MAT40	-	Parameter indicates the Moving Average Turnover over the past '*t*' trading days; where *t* = 20 and 40.	movavg (turn over, *m*) where, $m \in (20, 40)$
MSV10 MSV20 MSV40	-	Parameter indicates the deviation in the trade volume during the past '*t*' trading days; where *t* = 10, 20, and 40.	movstd (volume, *m*) where, $m \in (10, 20, 40)$
P0P9 P0P19 P0P39	-	Parameter refers to Momentum lag and is used to identify the trend in price movement.	$\dfrac{p_{t-m}}{p_t}$; where, $m \in (9, 19, 39)$

movavg: moving average; movstd: moving standard deviation; m_1, m_2: Holding period (in days).

Output Feature:

$$R^s_{t.m} = \frac{P^s_{t+m}}{P^s_t} - 1 \tag{1}$$

$$R^I_{t.m} = \frac{P^I_{t+m}}{P^I_t} - 1 \tag{2}$$

$$ER^s_{t.m} = R^s_{t.m} - R^I_{t.m} \tag{3}$$

4. Methodology

The following section describes the design framework, the different machine learning models adopted for the current study, and the evaluation metrics considered. Figure 1 depicts the design framework for the entire work. The data were collected and cleaned before dividing into training and testing sets. All the models were trained using the training set and validation of the trained models was performed using the testing data using different metrics shown in Figure 1.

4.1. Random Forest

Random Forest is a very famous supervised machine learning technique widely used in regression and classification problems. The technique is efficient to handle continuous variables for regression as well as categorical variables for classification. Random Forest employs ensemble methodology, and its operation is based on multiple decision trees. The algorithm uses the idea of Bagging or Bootstrap Aggregation. Bagging selects a random sample out of the dataset and each model is generated from the samples (Bootstrap samples) taken from the dataset with replacement. This method of row sampling with replacement is known as bootstrap. Each model is trained independently, and the final output is obtained by majority voting by a combination of individual output from every model. This line of involvement of the combination of outputs from different models and generation of the final output by majority voting is known as aggregation. In the case of regression, instead

of taking majority voting, the average or mean of all the results from each model is taken into consideration (Breiman 2001). A Random Forest predictor is calculated as

$$y^C(x) = \frac{1}{C} \sum_{i=1}^{C} P_i(x) \tag{4}$$

where x is the input, C is the number of trees and $P_i(x)$ is a single regression tree.
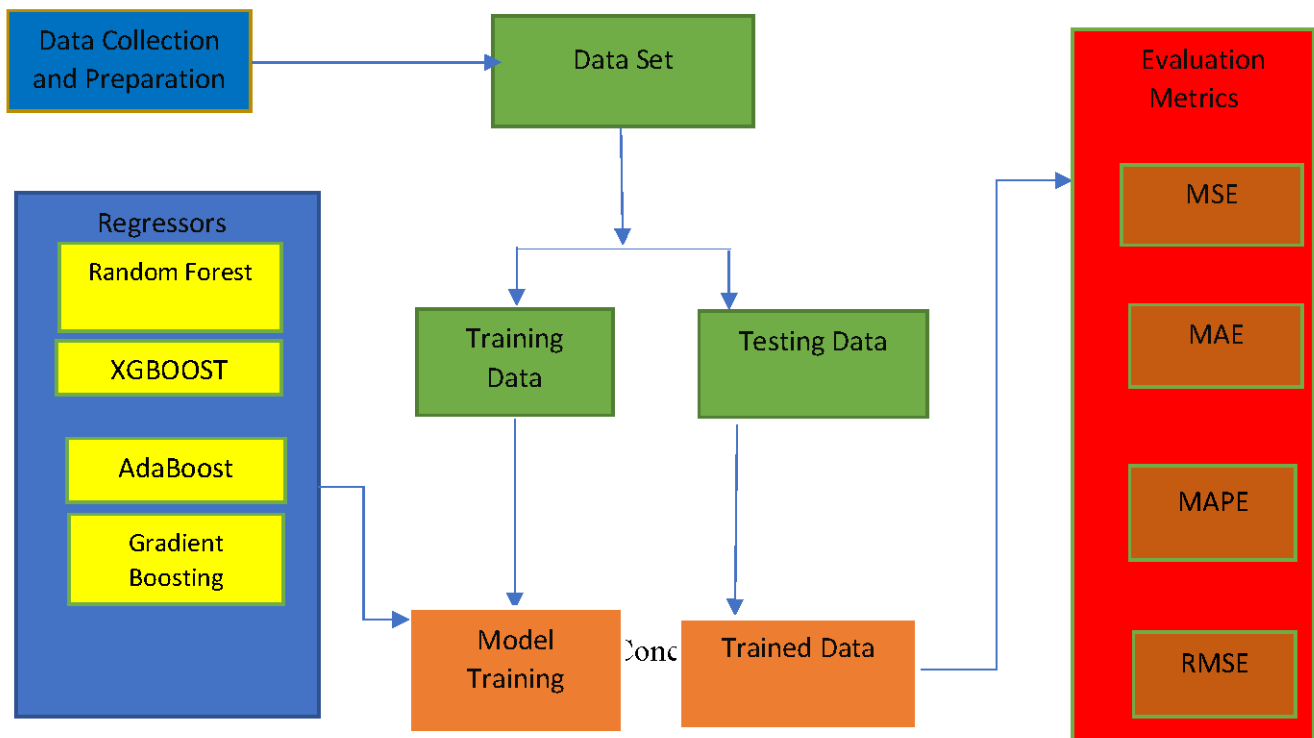
Figure 1. Feature Importance Plots for Different Models.

The calculation of Feature importance is done in accordance with the decrement in node impurity measured by the probability of reaching that node. The probability is the ratio of the samples reaching that node by the total number of samples. A higher value of probability implies more importance of the feature. For each decision tree (assuming a binary tree with two child nodes), node importance is calculated using Gini Importance. The equation is as follows:

$$k_i = w_i G_i - w_{left(i)} G_{left(i)} - w_{right(i)} G_{right(i)} \tag{5}$$

where k_i is the importance of node i, w_i is the weighted number of samples reaching node i, G_i is the impurity of node i, $left(i)$ and $right(i)$ denote the child note from the left split and right split of node i. The final feature importance of the Random Forest model is the average over all the individual trees. The summation of the value of feature importance on each tree is computed and further divided by the total number of trees.

The Random Forest algorithm can handle large datasets. However, when the data are very sparse, the model may face challenges as, for some nodes, the bootstrapped sample and the random subset of features may produce an invariant feature space. Overfitting is also a risk with Random Forest and should be given attention.

4.2. AdaBoost

Boosting is a theoretical concept conceived long before practical algorithms were developed based on the principle. The AdaBoost (adaptive boosting) algorithm is the first

algorithmic implementation based on the concept of boosting. Boosting is an ensemble learning technique that involves combining predictions from many weak learners. The idea is to create a model from the training data followed by another one that tries to work on the error generated by the initial model. Following it, the addition of the model is carried on till the data are predicted up to a satisfactory level. The AdaBoost algorithm combines the predictions from short one-level decision trees called decision stumps. Decision stumps are used as the algorithm keeps on adding weak models in order to correct their prediction. The model's weakness can be determined by the error rate of the weak estimator (Freund et al. 1999). Let us consider a sequence of n samples- $(x^1, y^1), (x^2, y^2), (x^3, y^3) \ldots (x^n, y^n)$. θ denotes the threshold for marking correct and incorrect predictions. The distribution is denoted by $D_p(i)$ for all i and $L(i)$ is the loss function. The loss function is calculated as

$$L(i) = |f(x_i) - y_i| \tag{6}$$

where $f(x_i)$ is the predicted output and y_i is the original output. The average loss is calculated as follows:

$$L_p = \sum_{i=1}^{n} L(i) D_p(i) \tag{7}$$

For AdaBoost regressor, the error rate of $f(x_i)$ is as follows:

$$\varepsilon = \sum D_p(i) \tag{8}$$

The model updates distribution D_p as

$$D_{p+1}(i) = \frac{D_p(i)}{Z} \times \left(\begin{array}{l} \beta \; if \; \left| \frac{f(x_i) - y_i}{y_i} \right| \leq \theta \\ 1 \; otherwise \end{array} \right) \tag{9}$$

where $\beta = \varepsilon^2$ and Z is a normalization factor.

AdaBoost works better in case of classification problems and the model is particularly vulnerable to uniform noise. It faces difficulty in adapting to different link functions to create a linear model with a given outcome.

4.3. Gradient Boosting

In Gradient Boosting, the statistical framework considers boosting to be an optimization problem in which the objective is to minimize the model's loss function by adding weak learners and employing a gradient descent type approach. The model considers stage-wise addition where a single weak learner is included at a time and existing learners of the models are left unchanged. The loss function must be differentiable and in regression problems, the squared error function is generally used. Decision trees are employed as weak leaners. More specifically, regression trees are employed where the outputs can be summed up and the following model outputs are further added to correct the residuals. The fundamental dissimilarity between AdaBoost and Gradient Boosting is the way the models detect the shortcomings-AdaBoost uses weights while Gradient Boosting employs a gradient descent approach to minimize the loss. One fine aspect of Gradient Boosting is its power of optimizing user-defined cost functions that are often problem-specific (Friedman 2002). The overall loss L_f in the model is given by

$$L_f = L(y, F_p(x)) + \alpha_p L(H(x, -\frac{dL}{dF_p(x)})) \tag{10}$$

In the above equation, p is the total number of samples, $F_p(x)$ is the actual target minus the updated predictions from all the models that have been built so far. The input and outputs are denoted by x and y. H represents the model. The gradient descent technique is

used to calculate the gradient term α_p such the loss function is minimized. The following equation will help to comprehend Equation (10).

$$F_{p+1}(x) = F_p(x) + \alpha_p H(x, e_p) \tag{11}$$

$$L = \left(y - F_p(x)\right)^2 \tag{12}$$

$$\frac{dL}{dF_p(x)} = -2(y - F_p(x)) \tag{13}$$

Gradient boosting works well in capturing complex patterns in the data as it generalizes the framework and allows for easier computation. It is not preferred in cases where there is no straightforward way to study how variables interact and contribute to the final prediction. Further, it is harder to tune than other models because of so many hyperparameters.

4.4. XGBoost

The XGBoost algorithm is an advancement of the Gradient Boosting model. The algorithm tries to minimize the problem of over-fitting and optimizes the computational complexity. The objective function is simplified to combine the predictive and regularization terms without compromising the computation speed. The entire input data are provided to the initial learner followed by the residuals, which are fed into the second learner with an intention to tackle the error of the weaker ones. The sum of predictions from all the learners generates the final one (Chen and Guestrin 2016). Considering the input variable x_i, at step t, the function for prediction is as follows:

$$p_i(t) = \sum_{q=1}^{t} p_q(x_i) = p_i(t-1) + p_t(x_i) \tag{14}$$

where the learner is presented as $p_t(x_i)$, the predictions at step and $t-1$ are represented $p_i(t)$ and $p_i(t-1)$.

The XGBoost model uses the following equation to measure the quality of the model.

$$Obj(t) = \sum_{q=1}^{n} l\left(y_{predicted}, y_{actual}\right) + \sum_{q=1}^{n} \wp(p_i) \tag{15}$$

where $l\left(y_{predicted}, y_{actual}\right)$ represents the training loss that measures how well the model fits on training data and $\wp(p_i)$ is the regularization factor measuring the complexity of the trees, and n is the total samples. Further, $\wp(p)$ can be expressed as

$$\wp(p) = \alpha T + \frac{1}{2}\beta\|w\|^2 \tag{16}$$

where the control parameter for regularization is presented by β, w is the score-vectors at the leaf nodes, α is the minimum loss for dividing a node.

XGBoost performs very well on medium and small datasets with subgroups and structured datasets with not too many features. XGBoost gives the best results in the case of structured or tabular datasets and is preferred in cases where computational speed is a great concern. It may not perform according to expectations in unstructured data related to computer vision and natural language processing.

4.5. Evaluating Metrics

The performance of the different prediction models is evaluated based on the following metrics: mean absolute error (MAE), mean squared error (MSE), mean absolute percentage

error (MAPE), root-mean-squared-error (RMSE). For a total of n number of samples, if y_i and p_i represents the actual and predicted values,

$$AE = \frac{1}{n} \sum_{i=1}^{n} |y_i - p_i| \tag{17}$$

$$MSE = \frac{1}{n} \sum_{i=1}^{n} (y_i - p_i)^2 \tag{18}$$

$$MAPE = \frac{1}{n} \sum_{i=1}^{n} \left| \frac{y_i - p_i}{y_i} \right| \tag{19}$$

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^{n} (y_i - p_i)^2} \tag{20}$$

5. Results

Four different models, namely, Random Forest, AdaBoost, Gradient Boosting, and XGBoost were employed in the current work. The implementations were carried on Google Colab which is a Jupyter notebook environment that runs entirely in the cloud. GPU runtime environments with Python v3.8.5 were used. For training and testing, we divided the whole dataset in the ratio of 80:20. To tune the hyperparameters, GridSearchCV, a function of sklearn's model_selection package was used for all models. For every model, five-fold cross-validation has been used.

Figure 2 represents the feature importance plots for the four different models. The figure depicts that the five most important features in Random Forest, Gradient Boosting, and XGBoost are the same although their order of importance differs. They are namely-MSV40, MAT40, POP39, MAT20, MSV20. The variables are already explained in Table 1. Only a difference is noted in AdaBoost where three factors, namely, MSV40, MAT40, AND MSV 20 are common among the top five features with respect to other models while POP19 AND 10by40 are the exceptions that are not present elsewhere among the top five. For Random Forest, the top seven features provide 75% of the information in the data with respect to the dependent variables. For all other boosting methods used, the top six features provided around 75% of the information. This information has been computed from the cumulative sum of the percentage importance generated by the tool.

Various error metrics like mean absolute error (MAE), mean squared error (MSE), mean absolute percentage error (MAPE), root-mean-squared-error (RMSE) have been used to check the performance of the models. Table 2 represents the details of the errors for different models. Considering the different metrics, the XGBoost model performs best among the four models considered in the present work followed by Random Forest. The MAE and RMSE values for XGBoost are 0.0327 and 0.0550 while that of Random Forest are 0.0353 and 0.0588. Even the MSE is also lowest for XGBoost. The max_depth that represents the depth of each tree in the forest is 90 for Random Forest and 60 for XGBoost algorithm. The n_estimators representing the number of trees in an ensemble model is 130 for Random Forest while it is 150 for XGBoost. The superior performance of the XGBoost model is due to regularized formalization to control over-fitting. The model attempts to minimize L1 and L2 objective functions that in turn introduces a function of convex loss and a penalty term for model complexity. The model keeps on involving new trees to predict the residuals of previous trees that are then merged with the previous ones. For the gradient boosting purpose in the current work, 'squared_error' is used as the loss. Following the Random Forest, the Gradient Boosting algorithm has also performed well.
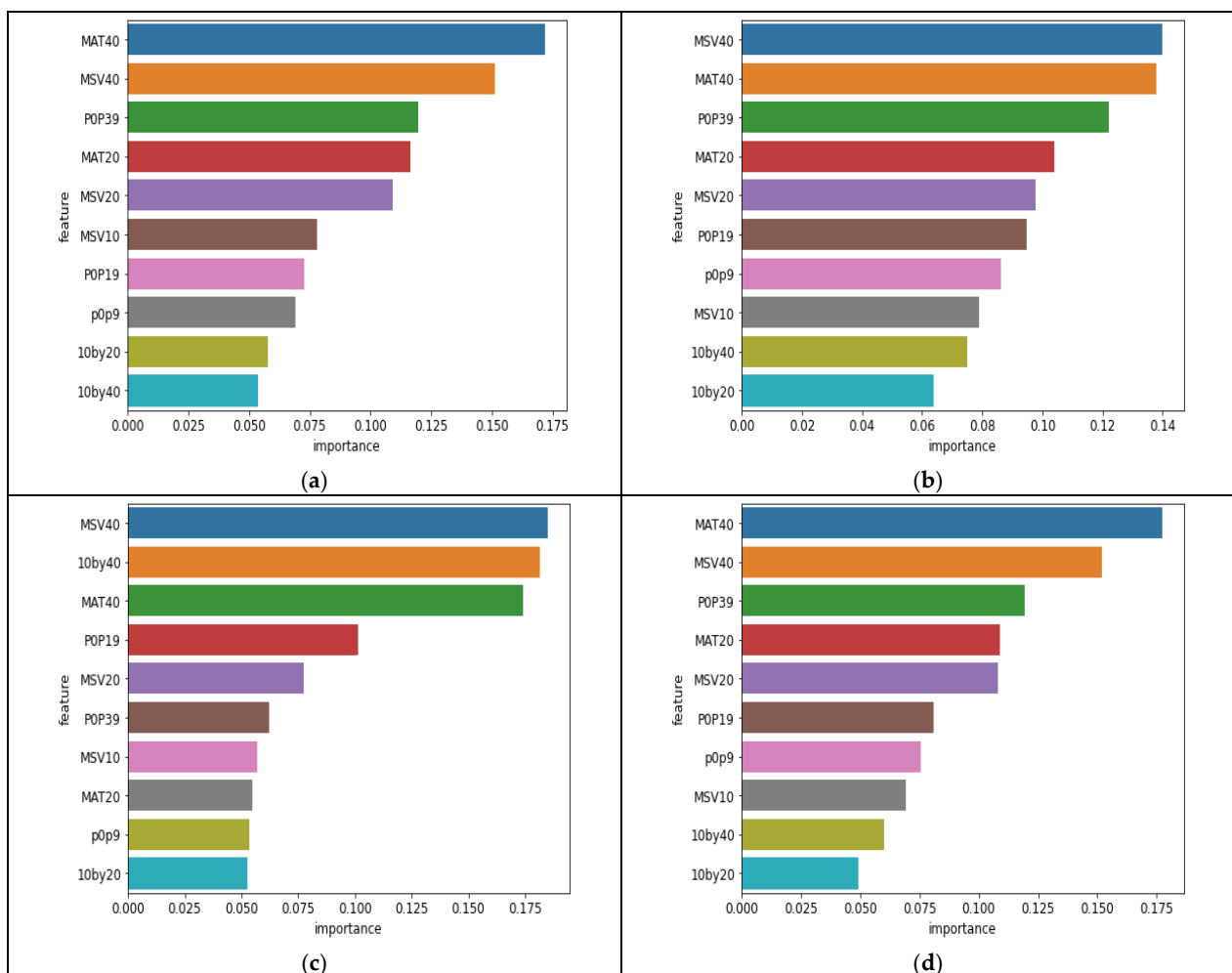
Figure 2. Feature Importance Plots for Different Models. (**a**) XGBoost. (**b**) Random Forest. (**c**) AdaBoost. (**d**) Gradient Boosting.

Table 2. Evaluation Metrics for Different Models.

	Random Forest	XGBOOST	AdaBoost	Gradient Boosting
MSE	0.0034	0.0030	0.0081	0.0041
MAE	0.0353	0.0327	0.0647	0.0423
MAPE	1.5644	1.7451	3.3125	2.2779
RMSE	0.0588	0.0550	0.0904	0.0645

To comprehend the prediction accuracy in a better manner, the predicted and actual values are plotted and demonstrated in Figure 3. These plots are one of the finest ways of data visualization to check the prediction performance of the models. The three models, namely, XGBoost, Random Forest, and Gradient Boosting perform in an appreciable manner as depicted in Figure 3 as most of the data in the scatter diagram are almost close to or on the diagonal line.
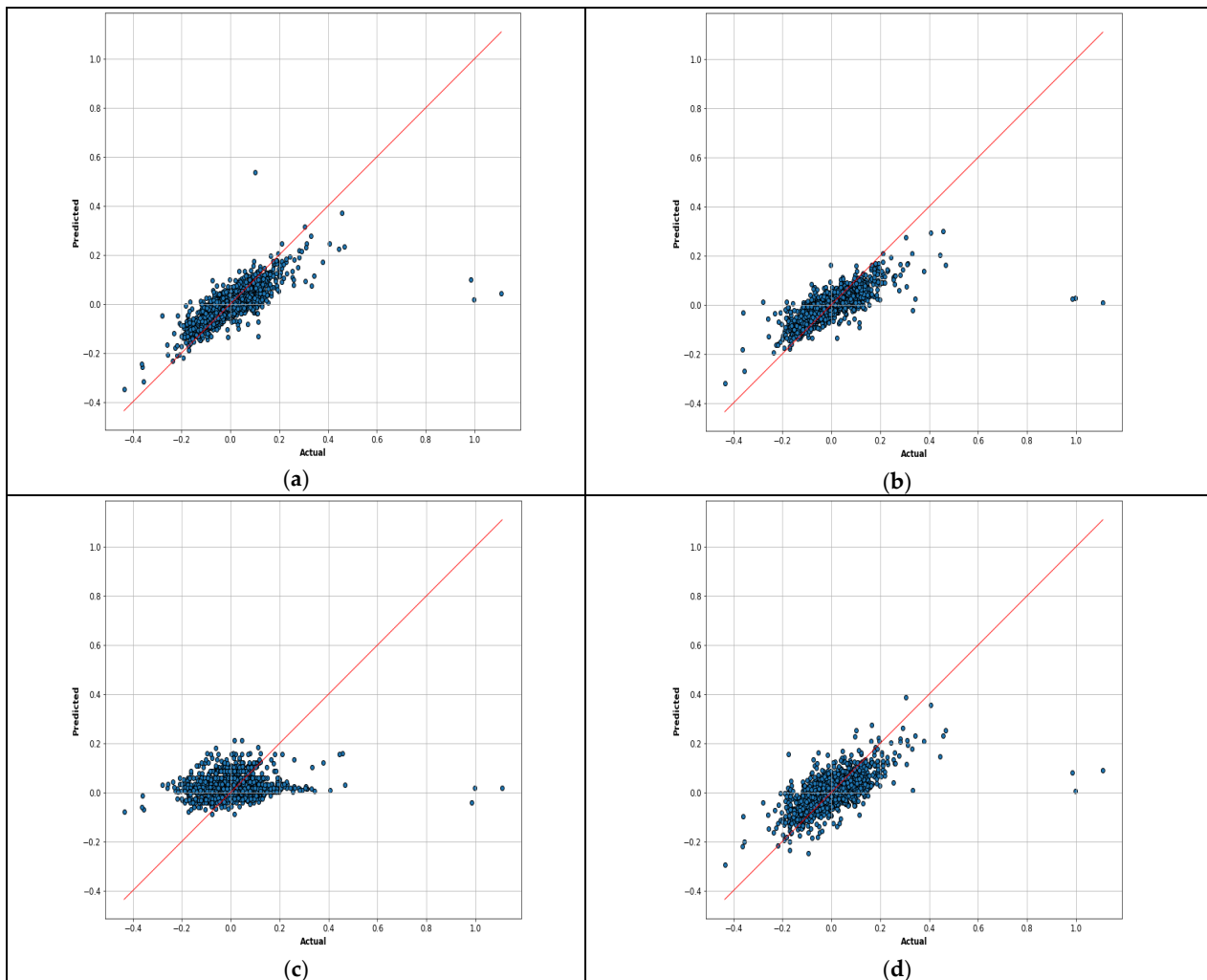
Figure 3. Actual versus Predicted Scatter Plots for Different Models. (**a**) XGBoost. (**b**) Random Forest. (**c**) AdaBoost. (**d**) Gradient Boosting.

With respect to other boosting, XGBoost can boost the weaker learner with better parallel computing and optimized algorithms. XGBoost has improved upon the base Gradient Boosting machine framework through systems optimization and algorithmic enhancements. The algorithm goes through a cycle. In the initial stage, it tests the existing models on a validation set; then, it further adds a model to improve the prediction accuracy followed by testing this new model along the existing models on a validation set again, and this cycle repeats till an optimal ensemble method is reached. The performance of the Random Forest model may be due to its ability to model non-linear dynamics in data. Further, the ensemble-based prediction technique incorporated into the Random Forest model contributes to its prediction accuracy. The individual trees in the forest protect each other from their error resulting in reduced cumulated error.

In brief, the proposed machine learning models help traders, investors, and portfolio managers better predict the stock market trends and in turn, the returns, particularly in the case of banking stocks, minimize their sole dependency on macroeconomic factors. The techniques further assist the market participants in pre-empting any price-volume action across stocks irrespective of their size, liquidity, or past turnover. Finally, the techniques are incredibly robust and display a strong capability in predicting trend forecasts, particularly with any large deviations. We further realize that given the multivariate predictive information presented by the models, a detailed in-depth robustness analysis of the machine learning techniques can be carried out across sectors to establish any

possible relationships over business cycles to improve the predictive performance as a future research gap.

6. Conclusions

The paper aims to validate the prowess of the different ensemble machine learning models in establishing the relevance of the technical indicators using the banking stocks in the Indian context. Our results indicate the significant predictive power of the machine learning technologies corresponding to the new technical indicators namely the price movement used to identify the trendlines, the standard deviation of the average trading volume over the past trading days, and the moving average turnover during the study period. We find the indicators to be highly significant across time horizons indicating that the relevance of the ensemble machine learning models stays put benefiting the short-term traders as well as the long-term investors. Applying the aforementioned technical indicators, we understand that the mean-variance investors might essentially deploy the ensemble machine learning methodologies for predicting the stock returns as well as for portfolio allocation. This evidence explains typically the predictive power of the machine learning technologies using the various technical indicators.

The present work explores the predictive power of the technical indicators, which we establish using an ensemble of machine learning techniques. We understand that existing asset pricing models that heavily rely on macroeconomic and sector-specific variables only partially explain the stock returns limiting their applicability and usage from the perspective of very short-term investors and intra-day traders. To overcome this pressing limitation, we use technical indicators as a relevant proxy for capturing the continuous trends in equity prices that are induced by the macro and related variables. In order to improve the predictive ability of these technical indicators, we deploy an ensemble of machine learning techniques for better decision-making. Our study aims to help the market participants by providing a data-based forecast that can demonstrate strong predictive potential. We believe that the predictability power of these ensemble techniques can be explored across sectors and asset classes that can be widely used equally by short-term investors as well as intra-day traders.

References

Ajmi, Ahdi Noomen, Shawkat Hammoudeh, Duc Khuong Nguyen, and Soodabeh Sarafrazi. 2014. How strong are the causal relationships between Islamic stock markets and conventional financial systems? Evidence from linear and nonlinear tests. *Journal of International Financial Markets, Institutions and Money* 28: 213–27. [CrossRef]

Alberg, John, and Zachary C. Lipton. 2017. Improving Factor-Based Quantitative Investing by Forecasting Company Fundamentals, (Nips). *arXiv* arXiv:1711.04837.

Ampomah, Ernest Kwame, Zhiguang Qin, and Gabriel Nyame. 2020. Evaluation of Tree-Based Ensemble Machine Learning Models in Predicting Stock Price Direction of Movement. *Information* 11: 332. [CrossRef]

Chatterjee, Ananda, Hrisav Bhowmick, and Jaydip Sen. 2021. Stock Price Prediction Using Time Series, Econometric, Machine Learning, and Deep Learning Models. Paper presented at 2021 IEEE Mysore Sub Section International Conference (MysuruCon), Hassan, India, October 24–25.

Andriyashin, Anton, Wolfgang K. Härdle, and Roman Vladimirovich Timofeev. 2008. Recursive Portfolio Selection with Decision Trees. SFB 649 Discussion Paper 2008-009. Available online: https://ssrn.com/abstract=2894287 (accessed on 15 January 2008). [CrossRef]

Ayala, Jordan, Miguel García-Torres, José Luis Vázquez Noguera, Francisco Gómez-Vela, and Federico Divina. 2021. Technical analysis strategy optimization using a machine learning approach in stock market indices. *Knowledge-Based Systems* 225: 107119. [CrossRef]

Belciug, Smaranda, and Adrian Victor Sandita. 2017. Business Intelligence: Statistics in predicting stock market. *University of Craiova—Mathematics and Computer Science Series* 44: 292–98.

Breiman, Leo. 2001. Random forests. *Machine Learning* 45: 5–32. [CrossRef]

Bustos, Oscar, and Alexandra Pomares-Quimbaya. 2020. Stock market movement forecast: A Systematic review. *Expert Systems with Applications* 156: 113464. [CrossRef]

Carlini, Federico, Doriana Cucinelli, Daniele Previtali, and Maria Gaia Soana. 2020. Don't talk too bad! stock market reactions to bank corporate governance news. *Journal of Banking & Finance* 121: 105962.

Challa, Madhavi Latha, Venkataramanaiah Malepati, and Siva Nageswara Rao Kolusu. 2020. S&p bse sensex and s&p bse it return forecasting using arima. *Financial Innovation* 6: 47.

Chang, Tsung-Sheng. 2011. A comparative study of artificial neural networks, and decision trees for digital game content stocks price prediction. *Expert Systems with Applications* 38: 14846–51. [CrossRef]

Chen, Tianqi, and Carlos Guestrin. 2016. XGBoost: A Scalable Tree Boosting System. Paper presented at KDD '16: The 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 13–17; pp. 785–94. [CrossRef]

Chen, Yufeng, Jinwang Wu, and Zhongrui Wu. 2022. China's commercial bank stock price prediction using a novel K-means-LSTM hybrid approach. *Expert Systems with Applications* 202: 117370. [CrossRef]

Cheng, Xu, Winston Wei Dou, and Zhipeng Liao. 2022. Macro-Finance Decoupling: Robust Evaluations of Macro Asset Pricing Models. *Econometrica* 90: 685–713. [CrossRef]

Chong, Eunsuk, Chulwoo Han, and Frank C. Park. 2017. Deep learning networks for stock market analysis and prediction: Methodology, data representations, and case studies. *Expert Systems with Applications* 83: 187–205. [CrossRef]

Chou, Jui-Shwng, and Thi-Kha Nguyen. 2018. Forward forecast of stock price using sliding-window metaheuristic-optimized machine-learning regression. *IEEE Transactions on Industrial Informatics* 14: 3132–42. [CrossRef]

Choudhry, Rohit, and Kumkum Garg. 2008. A hybrid machine learning system for stock market forecasting. *World Academy of Science, Engineering and Technology* 39: 315–18.

Ciner, Cetin. 2019. Do industry returns predict the stock market? A reprise using the random forest. *The Quarterly Review of Economics and Finance* 72: 152–58. [CrossRef]

Coyne, Scott, Praveen Madiraju, and Joseph Coelho. 2018. Forecasting stock prices using social media analysis. Paper presented at 2017 IEEE 15th International Conference on Dependable, Autonomic and Secure Computing, 15th International Conference on Pervasive Intelligence and Computing, 3rd International Conference on Big Data Intelligence and Computing and Cyber Science and Technology Congress (DASC/PiCom/DataCom/CyberSciTech), Orlando, FL, USA, November 6–10.

Dai, Zhifeng, Xiaodi Dong, Jie Kang, and Lianying Hong. 2020. Forecasting stock market returns: New technical indicators and two-step economic constraint method. *The North American Journal of Economics and Finance* 53: 101216. [CrossRef]

Day, Min-Yuh, Yensen Ni, Chinning Hsu, and Paoyu Huang. 2022. Do Investment Strategies Matter for Trading Global Clean Energy and Global Energy ETFs? *Energies* 15: 3328. [CrossRef]

De Oliveira, Fagner A., Cristiane N. Nobre, and Luis E. Zárate. 2013. Applying Artificial Neural Networks to prediction of stock price and improvement of the directional prediction index–Case study of PETR4, Petrobras, Brazil. *Expert Systems with Applications* 40: 7596–606. [CrossRef]

Dutta, Goutam, Pankaj Jha, Arnab Kumar Laha, and Neeraj Mohan. 2006. Artificial neural network models for forecasting stock price index in the Bombay stock exchange. *Journal of Emerging Market Finance* 5: 283–95. [CrossRef]

Edmans, Alex, Itay Goldstein, and Wei Jiang. 2012. *Feedback Effects and the Limits to Arbitrage*. Working Paper 17582. Cambridge: National Bureau of Economic Research.

Fama, Eugene F. 1970. Efficient capital markets: A review of theory and empirical work. *The Journal of Finance* 25: 383–417. [CrossRef]

Fama, Eugene F., and Kenneth R. French. 1993. Common risk factors in the returns on stocks and bonds. *Journal of Financial Economics* 33: 3–56. [CrossRef]

Fama, Eugene F., and Kenneth R. French. 2015. A five-factor asset pricing model. *Journal of Financial Economics* 116: 1–22. [CrossRef]

Freund, Yoav, Robert E. Schapire, and Naoki Abe. 1999. A short introduction to boosting. *Journal-Japanese Society for Artificial Intelligence* 14: 1612.

Friedman, Jerome H. 2002. Stochastic gradient boosting. *Computational Statistics & Data Analysis* 38: 367–78.

Guiso, Luigi, Paola Sapienza, and Luigi Zingales. 2008. Trusting the stock market. *Journal of Finance* 63: 2557–600. [CrossRef]

Guiso, Luigi, Paola Sapienza, and Luigi Zingales. 2011. *Time Varying Risk Aversion*. Working Paper. Evanston: Northwestern University.

Guresen, Erkam, Gulgun Kayakutlu, and Tugrul U. Daim. 2011. Using artificial neural network models in stock market index prediction. *Expert Systems with Applications* 38: 10389–97. [CrossRef]

Hanauer, Matthias X., Marina Kononova, and Marc Steffen Rapp. 2022. Boosting Agnostic Fundamental Analysis: Using Machine Learning to Identify Mispricing in European Stock Markets. *Finance Research Letters* 48: 102856. [CrossRef]

Hellström, Thomas, and Kenneth Holmström. 1998. *Predictable Patterns in Stock Returns*. Published as Opuscula ISRN HEV-BIB-OP-30-SE. Västerås: Center of Mathematical Modeling, Department of Mathematics and Physics, Mälardalen University, 31p.

Hsu, Ming-Wei, Stefan Lessmann, Ming-Chien Sung, Tiejun Ma, and Johnnie E. V. Johnson. 2016. Bridging the divide in financial market forecasting: Machine learners vs. financial economists. *Expert Systems with Applications* 61: 215–34. [CrossRef]

Hu, Hongping, Li Tang, Shuhua Zhang, and Haiyan Wang. 2018. Predicting the direction of stock markets using optimized neural networks with Google Trends. *Neurocomputing* 285: 188–95. [CrossRef]

Huang, Chin-Sheng, and Yi-Sheng Liu. 2019. Machine learning on stock price movement forecast: The sample of the Taiwan stock exchange. *International Journal of Economics and Financial Issues* 9: 189.

Kara, Yakup, Melek Acar Boyacioglu, and Ömer Kaan Baykan. 2011. Predicting direction of stock price index movement using artificial neural networks and support vector machines: The sample of the Istanbul Stock Exchange. *Expert Systems with Applications* 38: 5311–19. [CrossRef]

Kazem Ahad, Ebrahim Sharifi, Farookh Khadeer Hussain, Morteza Saberi, and Omar Khadeer Hussain. 2013. Support vector regression with chaos-based firefly algorithm for stock market price forecasting. *Applied Soft Computing* 13: 947–58. [CrossRef]

Khashei, Mehdi, and Mehdi Bijari. 2010. An artificial neural network (p, d, q) model for timeseries forecasting. *Expert Systems with Applications* 37: 479–89. [CrossRef]

Kim, Kyoung-jae, and Ingoo Han. 2000. Genetic algorithms approach to feature discretization in artificial neural networks for the prediction of stock price index. *Expert Systems with Applications* 19: 125–32. [CrossRef]

Kim, Sondo, Seungmo Ku, Woojin Chang, and Jae Wook Song. 2020. Predicting the direction of US stock prices using effective transfer entropy and machine learning techniques. *IEEE Access* 8: 111660–82. [CrossRef]

Kolarik, Thomas, and Gottfried Rudorfer. 1994. Time series forecasting using neural networks. *ACM Sigapl Apl Quote Quad* 25: 86–94. [CrossRef]

Krauss, Christopher, Xuan Anh Do, and Nicolas Huck. 2017. Deep neural networks, gradient-boosted trees, random forests: Statistical arbitrage on the S&P 500. *European Journal of Operational Research* 259: 689–702.

Kraus, Mathias, and Stefan Feuerriegel. 2017. Decision support from financial disclosures with deep neural networks and transfer learning. *Decision Support Systems* 104: 38–48. [CrossRef]

Kristjanpoller, Werner, and Kevin Michell. 2018. A stock market risk forecasting model through integration of switching regime, ANFIS and GARCH techniques. *Applied Soft Computing* 67: 106–16. [CrossRef]

Kumar, Manish, and M. Thenmozhi. 2007. Support vector machines approach to predict the S&P CNX NIFTY index returns. In *10th Capital Markets Conference, Indian Institute of Capital Markets Paper*. Maharashtra: Indian Institute of Capital Markets.

Lei, Lei. 2018. Wavelet neural network prediction method of stock price trend based on rough set attribute reduction. *Applied Soft Computing* 62: 923–32. [CrossRef]

Leung, Edward, Harald Lohre, David Mischlich, Yifei Shea, and Maximilian Stroh. 2021. The promises and pitfalls of machine learning for predicting stock returns. *The Journal of Financial Data Science* 3: 21–50. [CrossRef]

Majhi, Ritanjali, Ganapati Panda, and Gadadhar Sahoo. 2009. Development and performance evaluation of FLANN based model for forecasting of stock markets. *Expert Systems with Applications* 36: 6800–8. [CrossRef]

Meesad, Phayung, and Risul Islam Rasel. 2013. Predicting stock market price using support vector regression. Paper present at 2013 International Conference on Informatics, Electronics and Vision (ICIEV), Dhaka, Bangladesh, May 17–18; pp. 1–6.

Naik, Nagaraj, and Biju R. Mohan. 2020. Intraday stock prediction based on deep neural network. *National Academy Science Letters* 43: 241–46. [CrossRef]

Neely, Christopher J., David E. Rapach, Jun Tu, and Guofu Zhou. 2014. Forecasting the equity risk premium: The role of technical indicators. *Management Science* 60: 1772–91. [CrossRef]

Nti, Isaac Kofi, Adebayo Felix Adekoya, and Benjamin Asubam Weyori. 2020. A comprehensive evaluation of ensemble learning for stock-market prediction. *Journal of Big Data* 7: 20. [CrossRef]

Panda, Chakradhara, and V. Narasimhan. 2007. Forecasting exchange rate better with artificial neural network. *Journal of Policy Modeling* 29: 227–36. [CrossRef]

Park, Hyun Jun, Youngjun Kim, and Ha Young Kim. 2022. Stock market forecasting using a multi-task approach integrating long short-term memory and the random forest framework. *Applied Soft Computing* 114: 108106. [CrossRef]

Patel, Jigar, Sahil Shah, Priyank Thakkar, and K. Kotecha. 2015. Predicting stock and stock price index movement using trend deterministic data preparation and machine learning techniques. *Expert Systems with Applications* 42: 259–68. [CrossRef]

Qiu, Mingyue, and Yu Song. 2016. Predicting the direction of stock market index movement using an optimized artificial neural network model. *PLoS ONE* 11: e0155133. [CrossRef]

Qiu, Mingyue, Cheng Li, and Yu Song. 2016. Application of the Artificial Neural Network in predicting the direction of stock market index. Paper presented at 2016 10th International Conference on Complex, Intelligent, and Software Intensive Systems (CISIS), Fukuoka, Japan, July 6–8; pp. 219–23.

Rahman, Molla Ramizur, and Arun Kumar Misra. 2021. Bank Competition Using Networks: A Study on an Emerging Economy. *Journal of Risk and Financial Management* 14: 402. [CrossRef]

Rasekhschaffe, Keywan, and Robert Jones. 2019. Machine learning for stock selection. *Financial Analysts Journal* 75: 70–88. [CrossRef]

Sadorsky, Perry. 2021. A random forests approach to predicting clean energy stock prices. *Journal of Risk and Financial Management* 14: 48. [CrossRef]

Shen, Shunrong, Haomiao Jiang, and Tongda Zhang. 2012. *Stock Market Forecasting Using Machine Learning Algorithms*. Stanford: Department of Electrical Engineering, Stanford University, pp. 1–5.

Shetty, Shekar, Mohamed Musa, and Xavier Brédart. 2022. Bankruptcy Prediction Using Machine Learning Techniques. *Journal of Risk and Financial Management* 15: 35. [CrossRef]

Song, Yue-Gang, Yu-Long Zhou, and Ren-Jie Han. 2018. Neural networks for stock price prediction. *arXiv* arXiv:1805.11317.

Sorensen, Eric H., Keith L. Miller, and Chee K. Ooi. 2000. The decision tree approach to stock selection. *The Journal of Portfolio Management* 27: 42–52. [CrossRef]

Tsibouris, George, and Matthew Zeidenberg. 1995. Testing the efficient markets hypothesis with gradient descent algorithms. In *Neural Networks in the Capital Markets*. Hoboken: John Wiley & Sons.

Wang, Sheng, Yin Luo, Rochester Cahan, Miguel A. Alvarez, Javed Jussa, and Zongye Chen. 2012. Signal processing: The rise of the machines. *Deutsche Bank Quantitative Strategy*, June 5.

White, Halbert. 1988. Economic prediction using neural networks: The case of IBM daily stock returns. Paper presented at International Conference on Neural Networks (ICNN '88), San Diego, CA, USA, July 24–27; vol. 2, pp. 451–58.

Wu, Jimmy Ming-Tai, Zhongcui Li, Gautam Srivastava, Meng-Hsiun Tasi, and Jerry Chun-Wei Lin. 2021. A graph-based convolutional neural network stock price prediction with leading indicators. *Software: Practice and Experience* 51: 628–44. [CrossRef]

Zhang, Yuchen, and Shigeyuki Hamori. 2020. The predictability of the exchange rate when combining machine learning and fundamental models. *Journal of Risk and Financial Management* 13: 48. [CrossRef]

Zhao, Yang, Jianping Li, and Lean Yu. 2017. A deep learning ensemble approach for crude oil price forecasting. *Energy Economics* 66: 9–16. [CrossRef]

Zhang, Jin-Liang, Yue-Jun Zhang, and Lu Zhang. 2015. A novel hybrid method for crude oil price forecasting. *Energy Economics* 49: 649–59. [CrossRef]

Zhong, Xiao, and David Enke. 2017. A comprehensive cluster and classification mining procedure for daily stock market return forecasting. *Neurocomputing* 267: 152–68. [CrossRef]

Zhu, Min, David Philpotts, and Maxwell J. Stevenson. 2012. The benefits of tree-based models for stock selection. *Journal of Asset Management* 13: 437–48. [CrossRef]

Zhu, Min, David Philpotts, Ross Sparks, and Maxwell J. Stevenson. 2011. A hybrid approach to combining CART and logistic regression for stock ranking. *Journal of Portfolio Management* 38: 100–9. [CrossRef]