Predicting the Direction of Stock Market Prices Using Tree-Based Classifiers

Abstract

Predicting returns in the stock market is usually posed as a forecasting problem where prices are predicted. Intrinsic volatility in the stock market across the globe makes the task of prediction challenging. Consequently, forecasting and diffusion modeling undermines a diverse range of problems encountered in predicting trends in the stock market. Minimizing forecasting error would minimize investment risk. In the current work, we pose the problem as a direction-predicting exercise signifying gains and losses. We develop an experimental framework for the classification problem which predicts whether stock prices will increase or decrease with respect to the price prevailing n days earlier. Two algorithms, random forests, and gradient boosted decision trees (using XGBoost) facilitate this connection by using ensembles of decision trees. We test our approach and report the accuracies for a variety of companies as improvement over existing predictions. A novelty of the current work is about the selection of technical indicators and their use as features, with high accuracy for medium to long-run prediction of stock price direction.

Keywords: stock price movement, xgboost, random forests, machine classification

1. Introduction and Motivation

For a long time, it was believed that changes in the prices of stocks is not forecastable. The well known Random Walk hypothesis (Malkiel & Fama, 1970; Malkiel, 2003), and the Efficient Market Hypothesis (Jensen, 1978), which state that a market is efficient with respect to a current information set $I(t)$ if it is impossible to make economic gains in this market, led to this belief. In other words, if it is impossible to outperform the market owing to the randomness in stock prices, then unless a different (often excessive) type of risk is considered, economic profits cannot rise. It is unclear, however, how such risk would be measured (see the work by Timmermann & Granger (2004), where stock prices are treated as a martingale). Therefore, it should be of little doubt that predicting the trends in stock market prices is a challenging task and that there is high returns from improving value at risk forecasts (Halblieb & Pohlmeier, 2012). On the contrary, the Wisdom of Crowd hypothesis, which emerges from the theory of collaborative filtering, states that many individuals, each with limited information, can provide very accurate assessments if their information is elicited in an appropriate fashion. It is, however, not known to be useful for predicting stock market returns; nonetheless, some individual, as well as institutional investors are able to beat the market to make profits (Avery, Chevalier & Zeckhauser, 2016). The inefficiency of prediction gets accentuated due to various uncertainties involved and owing to the presence of multiple variables all of which can potentially influence the market value on a particular day. Over time, a number of explanatory variables have been added to this enormous literature (see a history of EMH in (Sewell, 2011; Beechey, Gruen & Vickery, 2000) etc.): these include country-specific economic conditions, investors' sentiments towards a particular company, political events, etc. Consequently, stock markets are susceptible to quick changes, which often turn into random fluctuations in the stock prices (for calibration of agent-based dynamics, refer to the work by Recchioni et al. (2015)). Notwithstanding, EMH has certain fault lines that should be mentioned before we develop two algorithms to observe the quality of predictability for stock prices.

Indeed, (Malkiel, 2003) offers a masterly discussion of the critique of the EMH, and suggests that, way back in 1973, he in general advised investors to purchase broad-based index funds that bought and held all the stocks in the market and that charged very low expenses. He admits that by the start of the twenty-first century, the intellectual dominance of the efficient market hypothesis had become far less universal. Many financial economists and statisticians began to believe that stock prices are at least partially predictable. Furthermore, a new breed of economists emphasized psychological and behavioral elements of stock-price determination and volatility (Christoffersen & Diebold, 2006), and came to believe that future stock prices are somewhat predictable on the basis of past stock price patterns as well as certain 'fundamental' valuation metrics. We have discussed this issue further while explaining how variables are chosen in atypical models of stock market prediction. Later, many economists made controversial claims that these predictable patterns enable investors to earn excess risk-adjusted rates of return. From Shiller's behavioral 'bandwagon effect' to 'head and shoulders' and 'double bottoms' formations in stock prices (Shiller, 2000), modest predictive power seems to exist (Lo, Mamaysky & Wang, 2000), nonetheless. (Malkiel & Fama, 1970) states that while the stock market may not be a mathematically perfect random walk, it is important to distinguish statistical significance from economic significance. The statistical dependencies giving rise to momentum are extremely small and are not likely to permit investors to realize excess returns. The anomalies with stock returns were often modeled in a context-specific manner and do not qualify as generalization.

It is well-known that stock market price series are generally dynamic, non-parametric, chaotic and noisy in nature making investments intrinsically risky. In addition, in view of the model specification to follow shortly, we are mindful of the fact that stock market price movement is considered to be a random process with fluctuations that are more prominent in the short-run. It is needless to mention that advanced knowledge of near future stock price movements should help in minimizing this risk. Traders are more likely to buy a stock in the current period whose value is expected to increase in the future and conversely for falling prices. It is straightforward therefore, that accurate prediction of the trends in stock market prices maximizes capital gains and minimizes losses. Thus, it had best be admitted that adding value to this complex and deeply researched topic is not easy, especially in view of the millions of data points that are being generated around the world for every time period under consideration. To this end, this paper presents the use of techniques of Machine Learning (ML) to predict stock prices at the level of a firm to get better insights into the accuracy of price movements. Notably, ML techniques used towards forecasting has a long history. In fact, the use of a classifier system (Beltrametti et al., 1997) is not new to this literature. However, more intensive use of this technique with large set of data points generated every time period is only natural (viz. Xu, *et. al.*, 2018 on risk management for portfolio investments). The two ML algorithms that we use are, namely, *random forests* (RF) and forests of *gradient boosted decision trees* (GBDT). Decision trees in RF, work by finding the best threshold. Based on this, the feature space is recursively split. In comparison, GBDT models approximate regressors to the training samples and find the best split of the aggregate of the regressor functions. In this connection, it is useful to mention that XGBoost is a fairly new invention, offering a tool that reduces the time taken to construct GBDTs by reducing the time for training of a complete model.

It is to be noted that, application of ML models in stock market behavior is a rather recent phenomenon (Khaidem, Saha & Dey, 2016). The approach is a departure from traditional forecasting and diffusion type methods. Standard models used in stock price forecasting involves statistical methods such as time series modeling and multivariate analysis (Gencay, 1999; Timmermann & Granger, 2004; Bao & Yang, 2008), where, the stock price movements are usually treated as a function of time and solved as a regression problem. Conversely, in this paper as we pose it as a classification problem, the class label of each sample is determined by considering the t-day return. In our analysis, we have conducted experiments on $t = 3, 5, 10, 15, 30, 60,$ and 90 days. The goal is to design an intelligent model that learns from the market data using machine learning techniques and predicts the *direction* in which a stock price will change at the closing time everyday. The ability to forecast direction of stock prices for individuals and companies capable of holding on to investments over medium to long-run should be a very useful support to this literature.

2. Related Work

As a relevant perspective, note that, stock market analysis and prediction have been studied with the aid of methods such as machine learning and text mining. The literature below shall highlight some of the papers that use various algorithms under ML techniques and applies it to a limited set of stocks, country-wise and company-wise to evaluate the strength of the propositions. The use of prediction algorithms to determine future trends in stock market prices (Widom, 1995; Hellstrom & Holmstromm, 1998; Gencay, 1999; Li, Yang & Li, 2014; Dai & Zhang, 2013; Timmermann & Granger, 2004; Bao & Yang, 2008) is a way to improve upon the predictive ability and to re-evaluate the efficient market hypothesis and diffusion models (Saha, Routh & Goswami, 2014). The debate is by no means over, since algorithms that can model more complex dynamics of the financial system (Malkiel, 2003) have added to the prevailing controversy about whether stock prices are at all predictable. In addition, researchers considered esoteric ideas in this regard, such as the correlation between volatility in the stock market and songs in the billboard top 100 (Maymin, 2012), and standard approaches such as event analysis (Khanal & Mishra, 2017) for analysis of stock price reactions to stock dividend announcements. Furthermore, since behavior and individualized responses play a significant role in dictating the stock turnovers and prices, a few studies have engaged with word analysis of news articles (Mittermayer, 2004; Nikfarjam, Emadzadeh, & Muthaiyah, 2010; Nyberg, 2011; Kim, Jeong, & Ghani, 2014) and its predictive ability. ML, however, is a set of techniques that are relatively new. Consequently, several algorithms have been used in stock prediction such as support vector machine (SVM), artificial neural networks (ANN), linear discriminant analysis (LDA), linear regression, K-NN, and naïve Bayesian Classifier (Khan et al., 2014) to approach the subject of predictability with greater accuracy.

Of these, the studies applying data mining commonly use daily stock data. For example, prediction studies based on support vector machines (SVMs) (Cao & Tey, 2001; Ince & Trafalis, 2007; Atsalakis & Valavanis, 2009) have been conducted to determine pattern categories. The relevant literature survey reveals that SVM has been used most of the time in stock prediction research. Typically, autoregressive integrated moving average (ARIMA) model (Pai & Lin, 2005; Wang & Leu, 1996; Moskowitz, Ooi, & Pedersen, 2012) are used for identifying and predicting time series variations. The sensitivity of stock prices to external conditions have duly been considered Li, Yang & Li (2014). These include, the external conditions that are taken into consideration covering daily quotes of commodity prices such as gold, crude oil, natural gas, corn, and cotton in two major foreign currencies (EUR, JPY). These studies also collected daily trading data of 2666 U.S stocks trading (or once traded) at NYSE or NASDAQ from 1^{st} January 2000 to 10^{th} November 2014. This dataset includes the opening price, closing price, highest price, lowest price, and trading volume of every stock for each day over which the data was collected. Features were derived using the information from the historical stock data as well as external variables which were mentioned earlier in this section. For these papers, logistic regression turned out to be the best model with a success rate of 55.65%. In the paper by Dai & Zhang (2013), the data used for the analysis were stock (closing) prices of the company 3M. The data contained daily stock information ranging from 1^{st} September 2008 to 11^{th} August 2013 (1471 data points). Multiple algorithms were chosen to train the prediction system. The algorithms used were logistic regression, quadratic discriminant analysis, and SVM. These algorithms were used for predicting the direction of the stock on the successive day corresponding to a given data sample; it also predicted the price after the next n days. The accuracy of the successive-day prediction model ranged from 44.52% to 58.2%. The results in (Dai & Zhang, 2013) were justified on the grounds that the US stock market is semi-strong efficient, meaning that, neither fundamental nor technical analysis can be used to achieve superior gains. However, the long-run prediction model produced better results which peaked when the time window was 44 days. SVM reported the highest accuracy of 79.3%. In Di (2014), the authors have used 3 stocks (AAPL, MSFT, AMZN) between 4^{th} January 2010 and 10^{th} December 2014. Various technical indicators such as RSI, On Balance Volume, Williams %R, etc. were used as features. Out of 84 features, an extremely randomized tree algorithm, as described by Geurts & Louppe (2014), was implemented for the selection of the most relevant features. These features were then fed to an SVM with RBF kernel for training. Devi (2015) has proposed a model which uses a hybrid cuckoo search with support vector machine (with Gaussian kernel): the cuckoo search method optimizes the parameters of support vector machine. The proposed model used technical indicators: RSI, Money Flow Index, EMA, Stochastic Oscillator, and MACD as features. The data used in the proposed system consists of daily closing prices of BSE-Sensex and CNX - Nifty from Yahoo finance between January 2013 and July 2014. Giacomel, Galante & Pareira (2015) proposed a trading agent based on a neural network ensemble that predicts if a certain stock is going to rise or fall. They evaluated their model using two datasets: the North American and the Brazilian stock markets and achieved hit rates of greater than 56%. They even performed a simulation based on the predictions of their classifier, whose results were promising. In other studies, Artificial Neural Networks (ANNs) (Kimoto et al., 1990; Kohara et al., 1997) have been employed to achieve good predictions even in the case of complex relationships of variables. Boonpeng & Jeatrakul (2015) implemented a one-against-all (OAA-NN) and one-against-one neural network (OAO-NN) to classify buy, hold, or sell data and compared their performance with a traditional neural network. Historical data of Stock Exchange of Thailand (SET) for seven years (3^{rd} January 2007 to 29^{th} August 2014) was selected for testing. It was found that OAA-NN performed better than OAO-NN and traditional NN models, producing an average accuracy of 72.50%. In (Qiu & Song, 2016), an optimized ANN using genetic algorithms (GA) has been tried to predict the direction of the stock market in a similar fashion of the current work; the hit ratio achieved here for two types of data, based on different sets of technical indicators, for the Nikkei 225 index (Tokyo Stock Exchange) are 61.87% and 81.27%. Clearly, the intense applications of ML have found wide acceptance all around and have arguably been successful in attaining reasonably good accuracy of predictions.

However, most of these studies have some limitations for medium to long-run predictions. First, without filtering for outliers, the predictions based on all historical data leads to potential errors. Second, although the total completion price is determined by a variety of factors such as the foreign purchase closing price and domestic selling completion amount, this set needs to be expanded in order to reduce omitted variables bias. Variables of importance may include, categories of financial ratios, macro, labour market and housing variables and measures of sentiment and leverage (Black et al., 2014; Cochrane, 2008). With respect to the current paper, it is important to note that the main purpose is to implement two distinct methods on stock data and highlight their advantage over other non-ensemble techniques within the machine learning approaches for analyzing and predicting stock prices. This does not warrant conducting a regression analysis. Therefore, our engagement with feature extraction and assigning of importance to respective variables will follow available wisdom, except that the outcomes are expected to be more efficient due to the choice of models.

The focus of the current paper is therefore to implement random forests, and gradient boosted trees on stock data, and to discuss its advantages over non-ensemble techniques. These models have been trained on time intervals of 3, 5, 10, 15, 30, 60, and 90 days days and the results are impressive. Moreover, majority of the related work focused on the time window of 10 to 44 days on an average as most of the previous studies preferred to use metric classifiers on time series data which was not smoothened. Therefore, these models are unable to learn from the data set when it comes to predicting for a long run window. In the method we propose, the data is first preprocessed using exponential smoothing, following which the increase or decrease in prices are calculated. After that, the classification algorithms are applied which do the job of predicting gains or losses. The extension of the time window to 90 days over which predictions are made is almost double the time adopted in previous studies (44 days). Significant improvement in accuracy obtained by our approach clearly establishes the superiority of the model developed.

It is important to note that forecasting techniques are rather insensitive to the crests and troughs in the time-series data. In the event of data point insufficiency or improper data conditioning, incorrect forecasting is common. The forecasting may generate some random value(s) which turns out to be outliers. For example, forecasting the price of a commodity as zero is one such instance. From an economic point of view, unless a commodity is a free good, usually publicly provided, the commodity price is unlikely to be zero in the market transactions. On the contrary, the classification model provides a probabilistic view of the predictive analysis and hence it plays a safer role as it predicts the *direction* of the trend. It uses the likelihood of the situation and hence the results are more trustworthy. A possible way to avoid the inherent problems in forecasting methods is therefore by recasting the problem as one that implements tree-based classifiers in ensemble learning as we adopt here. Our work takes a fresh and *gritty* perspective on the problem and our results are indicative of only potential gains or losses.

The remainder of the paper is organized as follows. Section 3 discusses the methods of preprocessing the classification, the features used (such as Relative Strength Index, Stochastic Oscillator, etc), and touches on how the experiments were performed. In Section 3.5, we discuss the dataset in detail. Section 4 discusses the results obtained, followed by a justification of the working of the method and a brief comparative analysis, establishing the superiority of the proposed system. We conclude by summarizing the results in Section 5. The results are further expanded in Appendix A.

3. Methodology and Analysis

In our experiments, the time series data acquired is first exponentially smoothed. Then the technical indicators are extracted. Technical indicators provide insights to the expected stock price behavior in future. These technical indicators are used as features to train the classifiers. The preprocessing, feature extraction, and classification methods are described in this section.

3.1. Data Preprocessing

Exponential smoothing grants larger weights to the recent observations and exponentially decreases weights of the past observations. The exponentially smoothed statistic of a series Y can be recursively calculated as:

$$
\begin{aligned}
S_0 &= Y_0 \\
\text{for } t > 0,\ S_t &= \alpha * Y_t + (1 - \alpha) * S_{t-1}
\end{aligned} \tag{1}
$$

where α is the smoothing factor and $0 < \alpha < 1$. Larger values of α reduce the level of smoothing. When $\alpha = 1$, the smoothed statistic becomes equal to the actual observation. The smoothed statistic S_t can be calculated as soon as consecutive observations are available. This smoothing removes random variation or noise from the historical data, allowing the model to easily identify the long-term price trend in the stock price behavior. Technical indicators are then calculated from the exponentially smoothed time series data which are later organized into a feature matrix. The target to be predicted in the i^{th} day is calculated as follows:

$$
target_i = sign(close_{i+d} - close_i) \tag{2}
$$

where d is the number of days after which the prediction is to be made. When the value of $target_i$ is +1, it indicates that there is a positive shift in the price after d days; -1 indicates that there is a negative shift after d days, giving us an idea of the direction of the prices for the respective stock. The $target_i$ values are assigned as labels to the i^{th} row in the feature matrix.

3.2. Feature Extraction from Data

In our solution, we consider only the closing price of a stock and we collect these values for many years. Hence, our input data can be considered to be of the form $(date, price_{closing})$. From the data, the following indicators are calculated:

1. **Relative Strength Index (RSI)**: RSI (Williams, 1978) is a popular momentum indicator which determines whether the stock is over-purchased or over-sold. A stock is said to be overbought when the demand unjustifiably pushes the price upwards. This condition is generally interpreted as a sign that the stock is overvalued and the price is likely to go down. A stock is said to be oversold when the price goes down sharply to a level below its true value. This is a result caused due to panic selling. RSI ranges from 0 to 100 and generally, when RSI is above 70, it may indicate that the stock is overbought and when RSI is below 30, it may indicate the stock is oversold.

2. **Stochastic Oscillator (SO)**: Stochastic Oscillator (Lane, 1984) follows the momentum of the price. As a rule, momentum changes before the price changes. It measures the level of the closing price relative to low-high range over a period of time.

3. **Williams Percentage Range (W%R)**: Williams Percentage Range (Williams, 1972) or Williams %R is another momentum indicator, similar in idea to stochastic oscillator. The Williams %R indicates the level of a market's closing price in relation to the highest price for the look-back period, which is 14 days. It's value ranges from -100 to 0. When its value is above -20, it indicates a *sell signal* and when its value is below -80, it indicates a *buy signal*.

4. **Moving Average Convergence Divergence (MACD)**: The moving average convergence-divergence (MACD) (Appel, 2005) is a momentum indicator which compares two moving averages of prices. The first moving average is a 26-day exponential moving average (EMA) and the second moving average is a 12-day EMA. The 26-day EMA is subtracted from the 12-day EMA. A 9-day EMA of the MACD is considered as the *signal line*, which serves as the threshold for the *buy* or *sell* signals.

5. **Price Rate of Change (PROC)**: The Price Rate of Change (PROC), (Larson, 2015) is a technical indicator which reflects the percentage change in price between the current price and the price over the window that we consider to be the time period of observation.

6. **On Balance Volume (OBV)**: On balance volume (OBV) (Granville, 1976) utilizes changes in volume to estimate changes in stock prices. This technical indicator is used to find buying and selling trends of a stock, by considering the cumulative volume: it cumulatively adds the volumes on days when the prices go up, and subtracts the volume on the days when prices go down, compared to the prices of the previous day.

3.3. Machine Learning Algorithms

Here we describe the algorithms that we have used for classification.

1. **Random Forest**: Decision trees (Quinlan, 1992; Geurts & Louppe, 2014) and random forests (Breiman, 2001) are popular machine learning approaches which can be used to solve a wide range of problems in classification. The basic training principle of decision trees is the recursive partitioning of the feature space using a tree structure, where each child node is split until pure nodes, i.e nodes which contain samples of a single class, are achieved. The splitting is done by the means of a criteria which tries to maximize the purity of the child nodes relative to their respective parent nodes. As maximum purity is ensured in child nodes, subsequently, pure nodes are arrived at. These pure nodes are not split further and constitute the leaf nodes. When a decision tree is used for the classification of a test sample, it is traced all the way down to a leaf node of the tree; as the leaf nodes of a decision tree are pure, the respective test sample is assigned the class label of the training samples of leaf node it arrives at. Random forests use an ensemble of many decision trees to reduce the effects of over-fitting. In a random forest, each tree is grown on a random subset of the feature space. Usually, if each sample in the data set has M features, $m = \sqrt{M}$ features are randomly selected to grow each tree. The reason random forests are preferred over decision trees is that random forests use a significant amount of voting based conclusions as compared to that of decision trees. It runs a bootstrap aggregation (or *bagging*) (Breiman, 2001) based routine by using a large number of de-correlated decision trees to classify a test sample. This course of operations is highly suitable for the stock data and its associated classification, as it meticulously examines the feature space to make better judgments over which class to finalize as the expected outcome.
   Gini impurity is used as the function to measure the quality of split in each node. Gini impurity at node N is given by:
   $$G(N) = 1 - (P_1)^2 - (P_{-1})^2 \tag{3}$$
   where P_i is the proportion of the population with class label i. The obvious heuristic approach to choose the best splitting decision at a node is the one that reduces the impurity as much as possible. In order words, the best split is characterized by the highest gain in information or the highest reduction in impurity.
   Random forests are *non-metric* classifiers, which means that unlike gradient-based methods, there are no learning-parameters which need to be set, and unlike Bayesian methods, it does not require an assumption of a prior distribution (Quinlan, 1992). Additionally, this is one of the reasons that has made random forests a popular classifier for various tasks.

2. **Gradient Boosted Decision Trees**: *Boosting* a classifier means combining the results of many weak predictors to make a strong prediction. Boosting has many variants but all of them work by optimally selecting or building successive weak classifiers such that the prediction resulting from many weak classifiers is strong. Gradient boosting is an improvement on decision trees, where each tree is approximated as an aggregate of many regressor functions $f_i(x)$. This is in contrast to the traditional idea of decision trees, wherein a Gini impurity based splitting is directly used to find the best split in the feature space. Each successive function f_i is built such that the misclassification rate successively decreases. This is done by trying to better classify the *residuals*, or the misclassified samples of the i^{th} iteration in the $i + 1^{th}$ iteration. Hence, the error in classification successively decreases. This step-wise aggregation of functions approximates a node in a tree, and eventually, the entire tree is approximated. Once each tree has been optimally approximated, the *structure scores* and *gain* are calculated, based on which the best split is determined. There are obvious advantages of doing this:

since each tree is built carefully, a lot of random trees need not be used in classifying a random sample, substantially decreasing the number of trees required in the forest of classifiers. XGBoost is a framework and library which parallelizes the growth of gradient boosted trees in a forest. The idea of gradient boosting (Friedman, 2000) comes from the principle of gradient descent: a greater number of the misclassified samples of the i^{th} learner should be classified correctly by the $i+1^{th}$ learner, and so on. This implies that the error in classification reduces as more number of regressors are constructed in each node. More specifically, in the case of gradient boosted decision trees (GBDT), the $i+1^{th}$ regression function is expected to rectify the mistakes of the i^{th} function. Since the error goes on decreasing as a tree is approximated by a larger number of functions, gradient boosting is considered to be a convex optimization problem. XGBoost (Chen & Guestrin, 2016) aims to minimize the time required to grow trees. This makes GBDTs more practical to use. Here, too, a subset of the features is used to build each tree. However, in the algorithmic description of XGBoost, the set of features being used to grow the tree is deliberately excluded: the assumption is that a subset of features is drawn out and fed into the algorithm.

3.4. Framework and Experimental Setup

The various aspects of the entire system consists of parts which need to be appropriately set. In all the experiments we performed, we used the following as settings for the preprocessing and classification:

1. The value of α for the exponential smoothing function was taken to be 0.095.
2. The trading window was varied as 3, 5, 10, 15, 30, 60, and 90 days.
3. The number of trees in random forests and XGBoost are taken to be 100. The more the number of trees, the better is the prediction, as proven by Chebyshev's inequality (see Section 2 of the Supplementary File).
4. In trees in random forests, one variable is used in each node for a split whereas as function is regressed on the variables in a node in a tree in gradient boosted trees. The Gini impurity criteria then decides which is the best split at a node.

The same settings are used for all the experiment runs for different time windows. The value of α is deliberately kept small in order to cause a small change the prices of the stocks. After the data is preprocessed and prepared for classification, we end up with a dataset containing the technical indicators as features and directions ± 1 as class labels. Following this, the dataset is randomly divided into training and test sets and for each trading window for each stock, the experiments are repeated many times over. The average of the results are presented in Section 4 and Appendix A.

3.5. Data Set & Features

The results are based on data for ten companies (see Appendix A) and all of the data available have been used, i.e., starting from the day they went public till 3^{rd} February 2017. These companies were sampled randomly, without any rigorous consideration of their background or the kind of economic impact they have on society. Some of these companies are software companies (FB, TWTR), electronics companies (AMS), automobile (TATA), etc. We like to emphasize that the diversity of the background of companies thus chosen for analysis of stock prices is crucial for ensuring the efficacy of the algorithms.

The raw values considered from the data include that acquired at the date of entry, the closing price, and the volume, etc. In their raw form, the size of the data corresponding to the stocks of different companies varied anywhere between 10kB and 700kB, with the number of rows (corresponding to closing prices) varying between 1180 and 10,700. Based on the closing prices, the remaining technical indicators (used as features for the learning algorithms) are determined. The data sets do not contain categorical and ordinal variables: all the feature values are continuous. The trends generally observed among most features is non-linear. This makes tree-based classifiers an attractive suite of algorithms for exploration. As a first insight, we present the feature importances as determined by the algorithms in Table 1. From this, we can see that it is seldom the case, especially for shorter trade windows, to be excessively influenced by a single feature or a small feature set. Combining the fact that we used a small number of features with this observation, we see that any need for dimensionality reduction is effectively superseded.

Table 1: Feature Importances: Sample data set

Company	Indicator	Trading Window (in number of days) and Corresponding Importance of Features (in %)						
		3	5	10	15	30	60	90
AAPL	RSI	17.98	17.34	17.49	16.88	16.19	13.81	12.66
	SO	14.23	13.51	12.82	12.43	11.61	10.3	9.41
	W%R	14.12	13.63	12.72	12.56	11.51	10.34	9.28
	MACD	17.8	17.89	18.2	18.16	17.7	16.48	16.34
	PROC	16.74	16.28	15.19	15.53	16	18.96	21.6
	OBV	19.12	21.36	23.57	24.43	26.99	30.11	30.72
FB	RSI	17.43	18.31	18.01	17.3	15.6	14.35	12.06
	SO	14.86	14.5	12.58	11.78	10.25	9.22	9.45
	W%R	14.8	13.13	12.34	11.64	11.42	9.28	10.77
	MACD	17.7	18.26	17.15	17.9	16.18	12.81	14.01
	PROC	16.17	15.77	15.55	16.11	17.19	22.44	22.13
	OBV	19.04	20.03	24.37	25.28	29.37	31.91	31.58

Typically, when the feature space has too many variables for confounding, dimensionality reduction becomes a necessity. The feature space needs to be broken into non-overlapping subspaces so that the learning algorithms are trained efficiently to discriminate between classes (feature dependent). However, in this particular class of problems, where the number of features aren't too many, such an exercise doesn't facilitate the computational efficiency of the machine learning algorithms we applied. However, in order to understand the contribution of features toward effective discrimination (as an academic exercise), we have included this. As expected, all the features/technical indicators are significant enough. However percentage contribution of OBV increases drastically with the increase in the size of the trading window (Please see table 1), in comparison to the other technical indicators. This is observed to be consistent across all stocks considered in our experiments.

4. Results and Discussion

Diversity, reputation and fiscal health in company profiles have been considered before choosing the stock data. Since we have considered stocks from corporations involved in information technology and enabled services, social media, electronics and instrumentation, manufacturing, and pharmaceuticals, the performance of the model accommodates variability. Besides, the model needs to be evaluated for its robustness. The *sign predictability* is related to the stocks' variance over time. Naturally, the gains or losses of stocks which are stable would be easier to predict than stocks that are relatively noisy. A lower variance in the data implies a better predictive capability of ML classifiers, and from an economic point of view, it accounts for the stability. The measures of performance that are used to evaluate the robustness of a binary classifier are accuracy, precision, recall (also known as sensitivity), specificity and the area under the curve (AUC) of the ROC curve. We have also included the Brier score for the classifiers (which is like the mean squared error)[1].

For demonstrating the efficacy of our approach, we present the results of two stocks: those of Apple and Facebook as a representative sample. The entire experiment was implemented on 10 different stocks, the results of which are elaborated in Appendix A.

4.1. Results of Random Forests

The results of classification of the samples from the dataset of stocks of Facebook and Apple are given in Table 2. In general, the accuracy increases as the width of the window is increased. Another important observation is that the F-score also increases with the increase in window-width. These results are presented in Table 2.

[1]For a complete description of all these measures of performance, refer to Section 1 of the Supplementary File

Table 2: Results of classification using random forests.

Company Name	Trading Window	Accuracy	Recall	Precision	Specificity	F-Score	Brier Score	AUC
AAPL	3	65.26	0.71	0.66	0.58	0.68	0.35	0.70
	5	72.55	0.78	0.73	0.67	0.75	0.28	0.79
	10	78.80	0.81	0.80	0.76	0.81	0.21	0.87
	15	82.01	0.85	0.82	0.78	0.84	0.17	0.90
	30	85.34	0.88	0.86	0.81	0.87	0.15	0.93
	60	90.44	0.93	0.90	0.86	0.92	0.10	0.96
	90	93.02	0.95	0.93	0.90	0.94	0.07	0.98
FB	3	67.59	0.72	0.69	0.62	0.71	0.30	0.71
	5	74.15	0.84	0.73	0.62	0.78	0.25	0.80
	10	81.39	0.90	0.81	0.69	0.85	0.19	0.89
	15	86.06	0.89	0.90	0.80	0.89	0.14	0.92
	30	89.89	0.95	0.90	0.80	0.92	0.11	0.94
	60	89.63	0.98	0.89	0.61	0.94	0.10	0.93
	90	94.76	0.98	0.96	0.72	0.97	0.06	0.94

Table 3: Results of classification using XGBoost.

Company Name	Trading Window	Accuracy	Recall	Precision	Specificity	F-Score	Brier Score	AUC
AAPL	3	55.99	0.80	0.56	0.29	0.66	0.44	0.57
	5	59.64	0.77	0.60	0.40	0.67	0.40	0.62
	10	61.50	0.79	0.61	0.41	0.69	0.39	0.66
	15	64.48	0.78	0.64	0.48	0.70	0.36	0.70
	30	68.82	0.85	0.68	0.47	0.76	0.31	0.76
	60	72.61	0.90	0.70	0.49	0.79	0.27	0.83
	90	77.13	0.88	0.77	0.61	0.82	0.23	0.86
FB	3	59.31	0.78	0.60	0.37	0.68	0.41	0.63
	5	65.05	0.79	0.65	0.48	0.71	0.35	0.72
	10	68.75	0.80	0.71	0.53	0.75	0.31	0.75
	15	80.84	0.86	0.85	0.71	0.86	0.19	0.86
	30	86.88	0.93	0.87	0.75	0.90	0.13	0.93
	60	88.76	0.98	0.89	0.57	0.93	0.11	0.89
	90	94.44	0.98	0.96	0.72	0.97	0.06	0.95

4.2. Results of XGBoost

The results of classification as well as the trends observed regarding the change in the classification accuracy and other metrics with the increase in the window-width in the case of XGBoost is similar to the trends observed in the case of random forests. Here, too, the classification accuracy and F-score increase with the increase in the window width. Moreover, the goodness of classification observed for a certain window-width in the case of XGBoost is comparable to the goodness of classification for the same window-width in the case of random forests. These results are presented in Table 3.

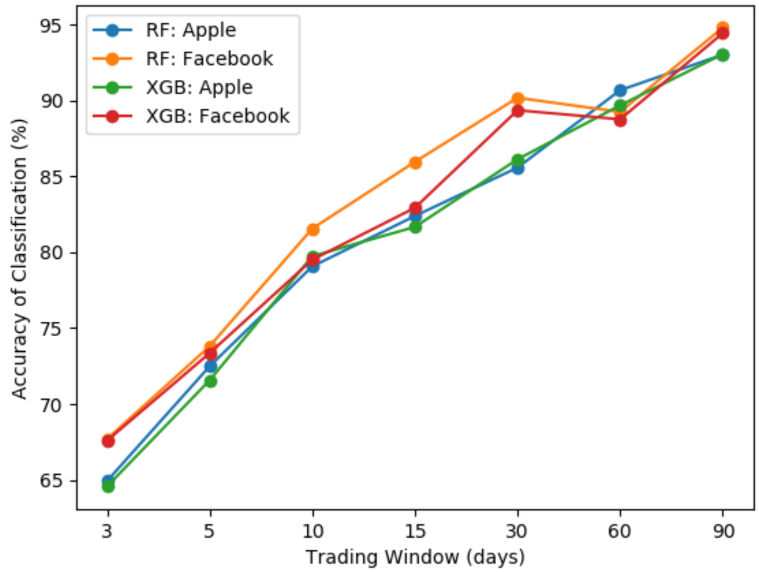

Figure 1: The trend of accuracy against the trading width considered. The accuracy of classification generally increases as the trading window increases for both random forests and XGBoost, used over the two datasets. The values are given in Tables 2 and 3.

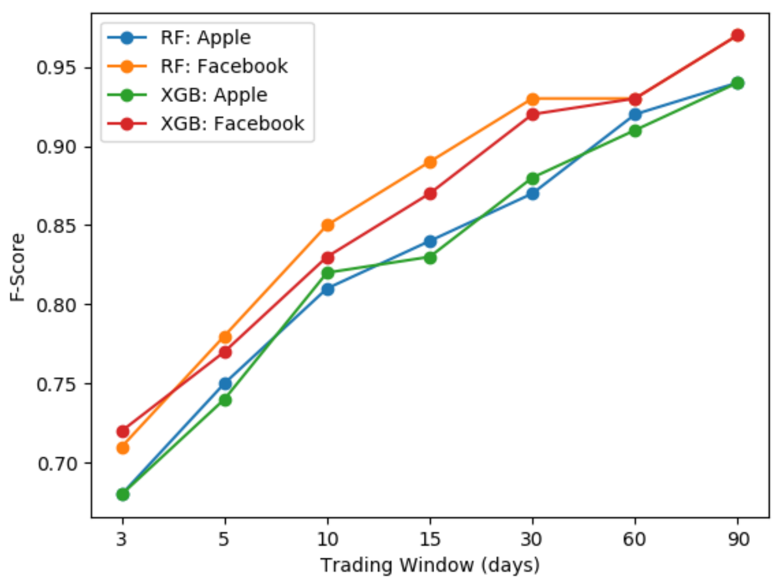

Figure 2: The trend of F-score against the trading width considered. The F-score increases as the trading window increases for both random forests and XGBoost, used over the two datasets. The values are given in Tables 2 and 3.

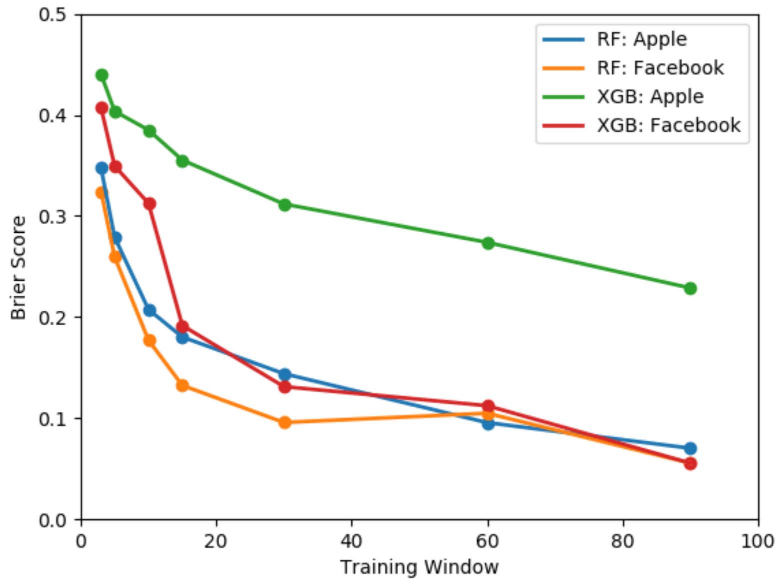

Figure 3: The trend of Brier score against the trading width considered. The Brier decreases as the trading window increases for both random forests and XGBoost, used over the two datasets. The values are given in Tables 2 and 3.

4.3. Comparison of Performance

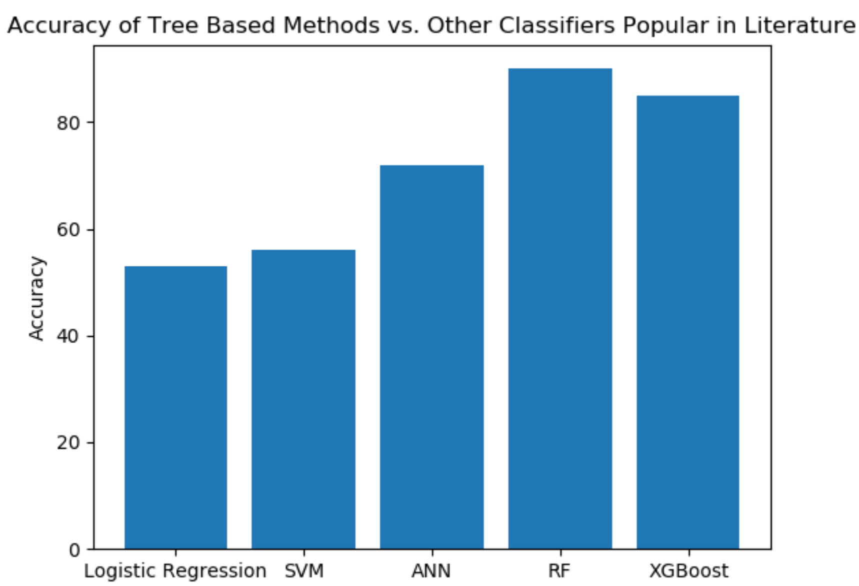

Figure 4: Comparison of the accuracy achieved with different classifiers. From the literature survey, it may be concluded that Logistic Regression performs poorly as compared to SVM (with a linear kernel), RF, and XGBoost. Also, from the bar graph, it is evident that SVM performs with an accuracy of less than 60% and RF and XGBoost outperforms SVM in terms of accuracy (close to 92% on an average, across different stocks). For a representative sample of ROC plots for SVM, refer to Section 7 of the supplementary file.

In this section, we compare the results of the classifiers previously used in literature (keeping the preprocessing steps the same) to compare the performance of random forests and GBDTs using XGBoost. We have implemented these classifiers and have reported the best-case average accuracy across different stocks for a performance baseline (please refer to Figure 4).

In the current work, we have used exponential smoothing to remove random local variation in the data. A detailed survey of the existing literature does not offer this as a preprocessing step to earlier predictions. The current methodology performs better than linear classifiers and the reason for this is the inherent nonlinearity in the data. In (Di, 2014), the authors have used a linear classifier as the supervised learning algorithm which yielded a highest accuracy of 55.65%. Our learning model to surpass all these metric classifiers in terms of long term prediction. An important practical question that is not entirely solved, is the criteria for the selection of the kernel function parameters – for Gaussian kernels the width parameter σ – and the value of ϵ in the ϵ loss insensitive function. Overall, the outcome of random forests and XGBoost are better than the classifiers previously tried and this is illustrated in Figure 4. Consistent with our goal, this is a general result implying a trend of performance across various stocks and is not pertaining to the stock of only one corporation.

4.4. Out-of-Bag (OOB) Error Visualization

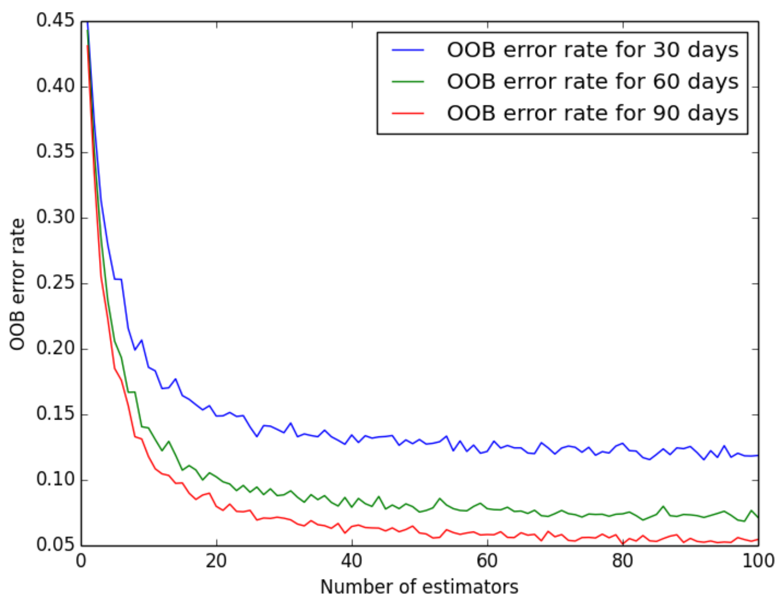

Figure 5: OOB error rate vs Number of estimators

After creating all the decision trees in the forest, for each training sample $Z_i = (X_i, Y_i)$ in the original training set T, we select all bagged sets T_k which does not contain Z_i. This set contains bootstrap datasets which do not contain a particular training sample from the original training dataset. These sets are called out of bags examples. There are n such sets for each n data samples in the original training dataset. OOB error is the average error for each Z_i calculated using predictions from the trees that do not contain it in their respective bootstrap sample. OOB error is an estimate of generalization error which measures how accurately the random forest predicts previously unseen data. We plotted the OOB error rate for our random forest classifier using the AAPL dataset.

From Figure 5, we can see that the OOB error rate decreases rapidly as more number of trees are added in the forest. However, a limiting value of the OOB error rate is reached eventually. The plot shows that the random forest converges as more number of trees are added in the forest. This result also explains why

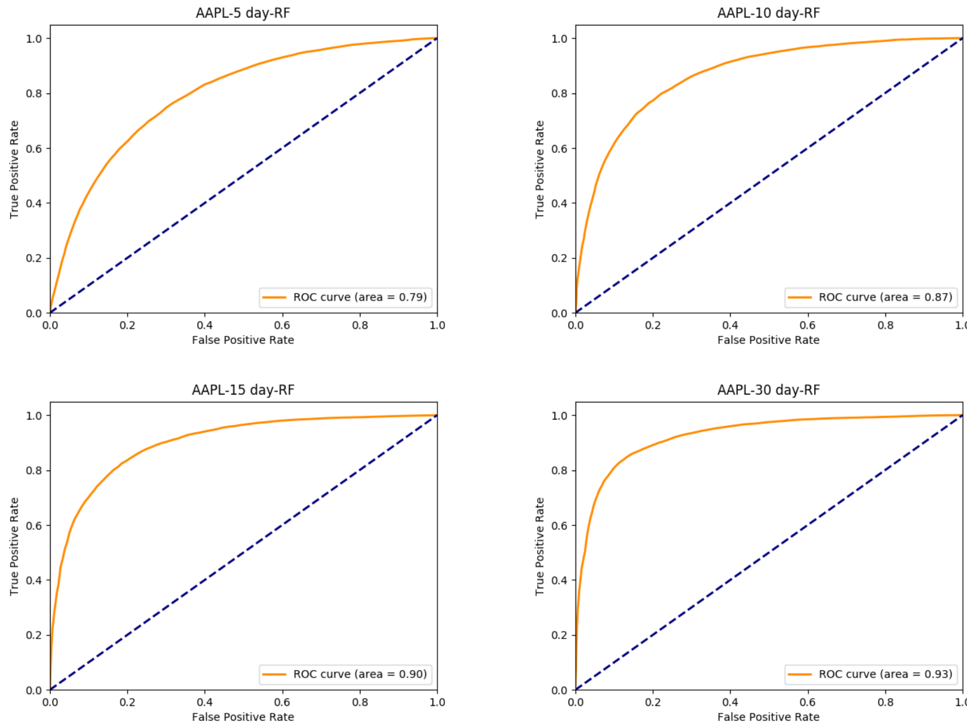

Figure 6: ROC curves plotted for random forests for 5, 10, 15, and 30 day trading windows. There is an improvement in performance with the increase in the trading window. For a more extensive list of ROC plots, refer to Section 5 of the Supplementary File.

random forests do not over-fit as more number of trees are added into the ensemble. A similar justification is for the efficacy of GBDTs.

4.5. Receiver Operating Characteristic Curves (ROC)

To analyze the effect of the duration of the trading window, we have plotted a small set of ROC curves in Figure 6. We observe that with an increase in the trading window, there is generally an improvement in classification accuracy. One reason why the accuracy improves with an increase in the value of t is that the economic indicators are able to capture more information regarding the movement of the prices over a larger time frame. This does not mean that the prices of a stock need to necessarily be increasing or decreasing, but just that with an adequate amount of information, an increase or decrease can be predicted accurately.

We have provided an exhaustive set of ROC plots in Section 5 of the Supplementary File.

4.6. Trading Indications: Inference From the Outcome of the Classifiers

Knowledge discovery from the analysis should create new frontiers or applications such as a trading strategy based on the strengths of the classification accuracy, investigating the behavior of certain classes of stocks. We achieved this as a derivative of the elaborate machine learning exercise.

In Figure 7, a subset of the trading suggestions by the random forest model is shown. The colored dots represent the trading decisions suggested by the model at data points with a 90-day time interval: the red dots suggest a drop in the prices and the blue dots suggest a rise in the prices in a time interval of every 90 days, and consequently, a suggestion of a purchase or sell. It can be seen from the graph that the model suggests to buy if it is predicted that price is going to rise after 90 days and the model suggests to sell if the price is predicted to fall after 90 days. It is an interesting question if the payoff function has a trend suggesting a trading model in time series. Notwithstanding, this is not an end-to-end trading strategy tool, but rather a component in a possibly more elaborate toolkit which can aid in investment decisions.

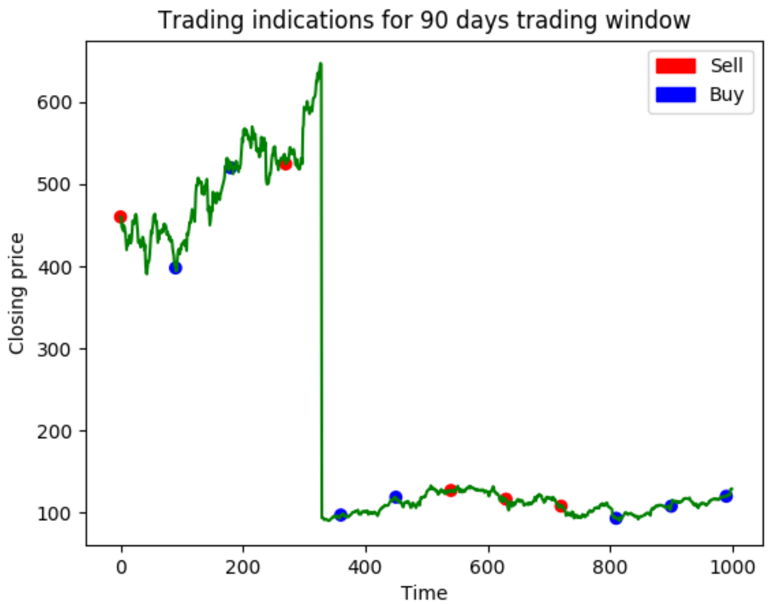

Figure 7: Trading suggestions made by the model on AAPL data. The X-axis is the n^{th} day in the dataset. The blue and red dots represent a small sample of trading suggestions made by the algorithm.

5. Conclusion

Application of machine learning techniques in stock price forecasting demands detailed execution, not only because of several technical complexities, but also because there are many deviations from best practice regulations in different countries. Consequently, analysis and forecasting of stock market activities and prices have often been country-specific or even region-specific allowing for variations in group behavior, culture, financial depth of the country and many other crucial determinants. The available approaches have made significant allowances for these variations and offered important results all along. However, there is room for improvement, where, the forecasting can be recast to bypass these drawbacks. In that sense, the proposed approach is a paradigm shift for available class of problems since it reformulates a traditional forecasting model as a classification problem. The present paper considers stock markets across places and industry-types and still offers a high accuracy for the predictive ability of the models developed - essentially in terms of whether stock prices are going to rise or fall over a period of time. The development of appropriate algorithms to represent big data through machine learning and subsequently, training these algorithms with real-life data involved considerable complexity in view of the well-known analytical and empirical contributions in this field. The paper has shown that the use of machine learning to understand the scope of big data is quite an advantage if models as developed here are applied to forecasting of the direction of stock prices and perhaps, for forecasting the degree to which stock prices change over time.

With reference to the directional prediction, the paper acknowledges that the high accuracy observed, could be a matter of concern. Natural suspicion about inherent bias in training data is common. However, the data set has been tested adequately in favor of non-existence of heavy bias. The absence of the effect of bias in the data has also been ascertained by examining the values of the F-scores in comparison to the accuracy (see Appendix A). Since the purpose of this paper has been to generalize and develop a method to predict with reasonable efficiency whether the price of a stock will increase or decrease on day $n+t$ as compared to day n, we argued that the mean accuracy is a more representative measure of the efficacy of the system as compared to the median returns. It has been discussed that in case of a positive drift in the prices of a stock it may be possible to have a better sign predictability, but that does not seem to have a conflict with our method of prediction. In addition, the data showed that most stock prices do not exhibit a strictly negative or a strictly positive drift. Therefore, the purpose of selecting stocks of companies like Nike, Toyota, Facebook, Amazon, Apple, etc. was to demonstrate that regardless of the background or domain of a company, and regardless of specific fluctuations in the prices over time, the efficacy of our methods hold and these do not suffer from diminishing accuracies.

In this paper, we have used random forests and XGBoost classifiers, as two useful algorithms to build our predictive model, which produced impressive results. The model is found to be robust in predicting the direction of stock movements. The robustness of our model has been evaluated by calculating various parameters such as accuracy, precision, recall, specificity, and F-score. For all the datasets we have used, we were able to achieve high accuracies for long-term predictions. The comparative analysis testifies the efficacy of our model as it outperforms the models discussed in the literature survey. In addition to that, a novelty of the current work is about the selection of technical indicators and their applications as features. As the background of the problem that has been solved here is primarily that of financial analysis, the flexibility with the use of various features, each with it's own interpretation has been useful.

The paper discusses the merit of using random forests and XGBoost systems as compared to other techniques used under non-ensemble procedures. Indeed, it has been shown in several studies that trained SVM applied on Korean stock market (Kim, 2003), or DT plus ANN applied on Taiwan market (Tsai and Wong, 2009) reach 56% and 67% accuracy overall. Compared to this the present techniques attain 78% accuracy. In addition, the reason for choosing random forests over decision trees is because random forests uses a significant amount of voting-based conclusions as compared to that of decision trees. It runs a bagging based routine by using a large number of de-correlated decision trees (consider growing forests in random fashion) to classify a predicted class. This course of operation is highly suitable for the stock data and associated classification, as it meticulously examines the feature space to make better judgments over which class to finalize as the expected outcome.

Our model can be used for devising new strategies for trading or to perform stock portfolio management, changing stocks according to trend prediction. The proposed model is indeed a novel way to minimize the risk of investment in stock market by predicting the returns of a stock more accurately than existing algorithms applied so far. In future, we could build boosted tree models to predict trends for short time windows. Ensembles of different machine learning algorithms can also be checked for its robustness in stock prediction. We also recommend exploration of the application of deep learning practices in Stock Forecasting involving learning weight coefficients on large, directed, and layered graphs.

6. References

Appel, G. (2005). Technical Analysis Power Tools for Active Investors. ISBN:0-13-147902-4.

Atsalakis, G. S., & Valavanis, K. P. (2009). Surveying stock market forecasting techniques – Part II: Soft computing methods.Expert Systems with Applications,36(3), 5932–5941.

Avery, C. N., Chevalier, J. A. & Zeckhauser, R. J. (2016). The CAPS Prediction System and Stock Market Returns. Review of Finance. 20 (4), 1363-1381.

Bao D. & Yang Z. (2008). Intelligent stock trading system by turning point confirming and probabilistic reasoning, Expert Systems with Applications, 34 (1), 620-627.

Beechey M., Gruen D. & Vickery J. (2000). The Efficient Market Hypothesis: A Survey, Research Discussion Paper. Economic Research Department. Reserve Bank of Australia.

Behner, P., Schwarting, D., Vallerien, S., Ehrhardt, M., Beever, C. & Rollmann, D. (2009). Pharmaceutical Companies in the Economic Storm Navigating from a Position of Strength. Technical Report: BooZ& Co Analysis.

Beltrametti, L, Fiorentini, R, Marengo, L and Tamborini, R (1997), A learning-to-forecast experiment on the foreign exchange market with a classifier system, Journal of Economic Dynamics and Control, Volume 21, Issues 8–9, 1543–1575.

Black, A. J., Klinkowska, O., McMillan, D. G. and McMillan, F. J. (2014), Predicting stock returns: Do commodities prices help?, Journal of Forecasting, 33, 627–639.

Boonpeng, S., & Jeatrakul, P. (2016). Decision Support System for Investing in Stock Market by using OAA-Neural Network. 8th International Conference on Advanced Computational Intelligence Chiang Mai, Thailand.

Breiman, L. (2001), Statistics Department, University of California Berkeley, CA 94720. Random Forests.

Bylander, T. & Hanzlik, D. (1999) Estimating Generalization Error Using Out-of-Bag Estimates. AAAI-99 Proceedings.

Cao, L. and Tay, F. E. (2001). Financial forecasting using support vector machines. Neural Computing & Applications, 10(2):184–192.

Chauhan, B., Umesh, B., Ajit, G., Sachin, K. (2014). Stock Market Prediction Using Artificial Neural Networks. International Journal of Computer Science and Information Technology. 5 (1), 904-907.

Chen, T., & Guestrin, C. (2016). XGBoost: A Scalable Tree Boosting System, Proceedings of the 22^{nd} ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD'16. doi:10.1145/2939672.2939785

Christoffersen, P. F., & Diebold, F. X. (2006). Financial asset returns, direction-of-change forecasting, and volatility dynamics. Management Science,52(8), 1273-1287.

Cochrane, J. (2008), The dog that did not bark: A defense of return predictability. Review of Financial Studies, 21, 1533-1575.

Dai, Y. & Zhang Y. (2013). Machine Learning in Stock Price Trend Forecasting. Stanford University, http://cs229.stanford.edu/proj2013/DaiZhang-MachineLearningInStockPriceTrendForecasting.pdf

Das, S. P. & Sudersan, P., Support Vector Machines for Prediction of Future Prices in Indian Stock Market. International Journal of Computer Applications. 41 (3), 22-26, 2012.

Devi, K. N., Bhaskaran, V. M. & Kumar, G. P. (2015). Cuckoo Optimized SVM for Stock Market Prediction, IEEE Sponsored 2nd International Conference on Innovations in Information, Embedded and Communication systems (ICJJECS).

Di, X. (2014), Stock Trend Prediction With Technical Indicators using SVM. Stanford University.

Friedman, J. H. (2000). Greedy Function Approximation: A Gradient Boosting Machine. Annals of Statistics. Vol. 29, p.1189–1232.

Gencay R. (1999). Linear, non-linear and essential foreign exchange rate prediction with simple technical trading rules. Journal of International Economics. Vol. 47, no.1, pp. 91-107.

Geurts, P., & Louppe, G. (2011). Learning to rank with extremely randomized tree. JMLR: Workshop and Conference Proceedings, 14, 49–61.

Giacomel ,F. , Galante, R. & Pareira, A. (2015). An Algorithmic Trading Agent based on a Neural Network Ensemble: a Case of Study in North American and Brazilian Stock Markets. IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology.

Granville, J. E. (1976). Granville's New Strategy of Daily Stock Market Timing for Maximum Profit. ISBN: 0-13-363432-9.

Halblieb, R. & Pohlmeier, W. (2012), Improving the value at risk forecasts: Theory and evidence from the financial crisis, Journal of Economic Dynamics and Control, Volume 36, Issue 8, 1212-1228.

Hellstrom, T. & Holmstromm, K. (1998). Predictable Patterns in Stock Returns. Technical Report Series IMA-TOM-1997-09.

Hunter, J. D. (2007) Matplotlib: A 2D graphics environment. Computing In Science & Engineering. 9(3), 90–95, DOI:10.1109/MCSE.2007.55.

Imandoust, S. B. & Bolandraftar, M., (2014). Forecasting the direction of stock market index movement using three data mining techniques: the case of Tehran Stock Exchange. International Journal of Engineering Research and Applications. 4(6), 106-117.

Ince, H. & Trafalis, T. B. (2007). Kernel principal component analysis and support vector machines for stock price prediction. IIE Transactions, 39(6):629–637.

Jensen M. C. (1978). Some Anomalous Evidence Regarding Market Efficiency. Journal of Financial Economics. 6 (2), 95-101.

Khaidem L., Saha S. & Roy Dey S. (2016). Predicting the direction of stock market prices using random forest. arXiv preprint arXiv:1605.00003.

Khan, W., Ghazanfar, M. A., Asam, M., Iqbal, A., Ahmed, S. & Khan, J. A. (2016). Predicting Trend In Stock Market Exchange Using Machine Learning Classifiers. Sci.Int, 28 (2), 1363-1367.

Khanal, A., R., Mishra, A., K. (2017). Stock price reactions to stock dividend announcements: A case from a sluggish economic period. The North American Journal of Economics and Finance. Volume 42, 338-345. doi.org/10.1016/j.najef.2017.08.002

Kim, Y., Jeong, S. R., & Ghani, I. (2014). Text opinion mining to analyze news for stock market prediction. Int. J. Advance. Soft Comput. Appl, 6(1).

Kimoto, T., Asakawa, K., Yoda, M., and Takeoka, M. (1990). Stock market prediction system with modular neural networks. In Neural Networks, 1990., 1990 IJCNN International Joint Conference on, pages 1–6. IEEE.

Kohara, K., Ishikawa, T., Fukuhara, Y., and Nakamura, Y. (1997). Stock price prediction using prior knowledge and neural networks. Intelligent systems in accounting, finance and management, 6(1):11–22.

Lane, G. (1984) "Lanes Stochastics. Second issue of Technical Analysis of Stocks and Commodities magazine. pp 87-90.

Ed. by Mark Larson (2015). Price Rate of Change: 12 Simple Technical Indicators: That Really Work, Wiley; https://doi.org/10.1002/9781119204428.ch5

Li, H., Yang, Z. & Li, T. (2014). Algorithmic Trading Strategy Based On Massive Data Mining. Stanford University. http://cs229.stanford.edu/proj2014/Haoming.

Lo, Andrew W., Mamaysky, H. & Wang, J. (2000). Foundations of Technical Analysis: Computational Algorithms, Statistical Inference, and Empirical Implementation, Journal of Finance, 55, 1705-1765.

Malkiel, B. G. (2003). The efficient market hypothesis and its critics. The Journal of Economic Perspectives. 17 (1), 59-82.

Malkiel, B. G. & Fama, E. F. (1970). Efficient capital markets: A review of theory and empirical work. The Journal of Finance. 25 (2), 383-417, 1970.

Mayankkumar, S. R. Y. & Patel, B. (2014). Stock prediction using artificial neural network. International Journal of Innovative Research in Science, Engineering and Technology. 3 (6), 76-84.

Maymin, P. (2012). Music and the market: Song and stock volatility. The North American Journal of Economics and Finance 23 (1), 70-85. doi.org/10.1016/j.najef.2011.11.004

Mittermayer, M. A. (2004). Forecasting intraday stock price trends with text mining techniques. In System Sciences, 2004. Proceedings of the 37th Annual Hawaii International Conference on, pages 30064.2–. IEEE.

Moskowitz, T. J., Ooi, Y. H., & Pedersen, L. H. (2012). Time series momentum.Journal of financial economics, 104(2), 228-250.

Nelson, D. M. Q., Pereira, A. C. M. & de Oliveira, R. A. (2017). Stock market's price movement prediction with LSTM neural networks. International Joint Conference on Neural Networks (IJCNN), 1419-1426.

Nikfarjam, A., Emadzadeh, E., and Muthaiyah, S. (2010). Text mining approaches for stock market prediction. In Computer and Automation Engineering (ICCAE), 2010 The 2nd International Conference on, volume 4, pages 256–260. IEEE.

Nyberg, H. (2011). Forecasting the direction of the US stock market with dynamic binary probit models.International Journal of Forecasting,27(2), 561-578.

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

Pedregosa F. et al. (2011). Scikit-learn: Machine Learning in Python. Journal of Machine Learning Research, 12, 2825–2830, 2011.

Recchioni, M, Tedeschi, G and Gallegati, M (2015), A calibration procedure for analyzing stock price dynamics in an agent-based framework, Journal of Economic Dynamics and Control, Volume 60, 1-25.

Qiu, M. & Song, U. (2016). Predicting the Direction of Stock Market Index Movement Using an Optimized Artificial Neural Network Model, PLOS ONE 11 (5).

Quinlan, J. R. (1986). Induction of Decision Trees, MACH. LEARN 1, pp 81–106

Saha, S., Routh, S. & Goswami, B. (2014). Modeling Vanilla Option prices: A simulation study by an implicit method. Journal of Advances in Mathematics. 6 (1), 834-848.

Sewell M. (2011). History of the Efficient Market Hypothesis. Research Note RN/11/04- UCL Dept. of Computer Science.

Shiller, Robert (2000), Irrational Exuberance, Princeton: Princeton University Press.

Timmermann, A. & Granger, C. W. (2004). Efficient market hypothesis and forecasting. International Journal of Forecasting. 20 (1), 15-27

Wang, J. H. & Leu, J. Y. (1996). Stock market trend prediction using arima-based neural networks. In Neural Networks, 1996., IEEE International Conference on, volume 4, pages 2160–2165. IEEE.

Widom, J. (1995). Research problems in data warehousing. In Proceedings of the fourth international conference on information and knowledge management. CIKM '95 (pp. 25- 30). New York, NY, USA: ACM. 10.1145/221270.221319.

Williams L. R. (1972). New Concepts in Technical Trading Systems. ISBN 978-0-89459-027-6.

Williams, L. R. (1978). The Secret of Selecting Stocks for Immediate and Substantial Gains. First Edition. ISBN 978-0930233051.

Xu, W, Chen, Y, Coleman, C and Coleman, T (2018), Moment matching machine learning methods for risk management of large variable annuity portfolios, Journal of Economic Dynamics and Control, Volume 87, 1-20.

Appendix A. Results

In this appendix, we elaborate the experimental results achieved by performing the experiments.

Table A.4: RF Results Table 1: results of random forests implemented on stocks of AAPL, AMS, AMZN, FB, and MSFT.

Company Name	Trading Window	Accuracy	Recall	Precision	Specificity	F-Score	Brier Score	AUC
AAPL	3	65.26	0.71	0.66	0.58	0.68	0.35	0.70
	5	72.55	0.78	0.73	0.67	0.75	0.28	0.79
	10	78.80	0.81	0.80	0.76	0.81	0.21	0.87
	15	82.01	0.85	0.82	0.78	0.84	0.17	0.90
	30	85.34	0.88	0.86	0.81	0.87	0.15	0.93
	60	90.44	0.93	0.90	0.86	0.92	0.10	0.96
	90	93.02	0.95	0.93	0.90	0.94	0.07	0.98
AMS	3	65.88	0.73	0.67	0.57	0.70	0.34	0.71
	5	68.80	0.74	0.68	0.63	0.71	0.31	0.76
	10	75.44	0.77	0.74	0.74	0.75	0.24	0.83
	15	79.60	0.79	0.78	0.80	0.78	0.21	0.87
	30	83.52	0.82	0.85	0.85	0.83	0.16	0.91
	60	89.24	0.90	0.89	0.89	0.89	0.11	0.95
	90	90.47	0.90	0.91	0.91	0.90	0.09	0.96
AMZN	3	67.09	0.75	0.66	0.58	0.70	0.33	0.72
	5	72.49	0.75	0.73	0.70	0.74	0.27	0.78
	10	77.43	0.81	0.77	0.74	0.79	0.23	0.86
	15	80.35	0.86	0.79	0.74	0.82	0.19	0.89
	30	86.48	0.92	0.86	0.79	0.89	0.13	0.93
	60	89.44	0.94	0.89	0.82	0.91	0.10	0.96
	90	94.79	0.96	0.96	0.93	0.96	0.05	0.98
FB	3	67.59	0.72	0.69	0.62	0.71	0.30	0.71
	5	74.15	0.84	0.73	0.62	0.78	0.25	0.80
	10	81.39	0.90	0.81	0.69	0.85	0.19	0.89
	15	86.06	0.89	0.90	0.80	0.89	0.14	0.92
	30	89.89	0.95	0.90	0.80	0.92	0.11	0.94
	60	89.63	0.98	0.89	0.61	0.94	0.10	0.93
	90	94.76	0.98	0.96	0.72	0.97	0.06	0.94
MSFT	3	67.30	0.72	0.67	0.62	0.70	0.33	0.73
	5	72.68	0.76	0.74	0.69	0.75	0.28	0.80
	10	77.39	0.85	0.76	0.68	0.80	0.23	0.85
	15	81.66	0.87	0.82	0.74	0.85	0.18	0.89
	30	86.15	0.88	0.88	0.84	0.88	0.14	0.93
	60	90.14	0.92	0.91	0.88	0.92	0.10	0.96
	90	92.24	0.94	0.93	0.90	0.93	0.08	0.97

Table A.5: RF Results Table 2: results of random forests implemented on stocks of NKE, SNE, TATA, TWTR, and TYO.

Company Name	Trading Window	Accuracy	Recall	Precision	Specificity	F-Score	Brier Score	AUC
NKE	3	66.22	0.72	0.68	0.59	0.70	0.33	0.72
	5	71.94	0.79	0.72	0.64	0.75	0.28	0.79
	10	76.01	0.83	0.76	0.67	0.79	0.25	0.84
	15	79.93	0.87	0.80	0.70	0.83	0.21	0.88
	30	85.76	0.93	0.85	0.74	0.89	0.15	0.93
	60	89.88	0.93	0.90	0.85	0.92	0.11	0.96
	90	93.31	0.96	0.93	0.88	0.95	0.06	0.98
SNE	3	63.53	0.67	0.63	0.60	0.65	0.37	0.69
	5	70.04	0.71	0.71	0.69	0.71	0.30	0.76
	10	76.90	0.79	0.77	0.75	0.78	0.23	0.85
	15	80.19	0.83	0.79	0.77	0.81	0.19	0.88
	30	83.45	0.84	0.84	0.83	0.84	0.17	0.91
	60	89.16	0.91	0.89	0.87	0.90	0.11	0.95
	90	89.75	0.92	0.89	0.88	0.90	0.11	0.96
TATA	3	67.29	0.62	0.66	0.72	0.64	0.32	0.74
	5	74.46	0.69	0.74	0.79	0.71	0.25	0.81
	10	79.78	0.73	0.81	0.86	0.76	0.21	0.87
	15	83.57	0.75	0.86	0.90	0.80	0.16	0.91
	30	87.64	0.83	0.89	0.92	0.86	0.13	0.94
	60	91.64	0.87	0.92	0.95	0.89	0.08	0.97
	90	95.44	0.94	0.94	0.96	0.94	0.04	0.99
TWTR	3	76.04	0.74	0.75	0.78	0.75	0.24	0.80
	5	75.53	0.78	0.69	0.74	0.73	0.24	0.83
	10	83.62	0.83	0.81	0.84	0.82	0.17	0.92
	15	85.82	0.89	0.79	0.83	0.84	0.14	0.93
	30	87.25	0.81	0.84	0.91	0.83	0.12	0.94
	60	87.82	0.78	0.83	0.93	0.80	0.12	0.94
	90	86.10	0.83	0.83	0.88	0.83	0.15	0.95
TYO	3	66.96	0.54	0.63	0.77	0.58	0.32	0.71
	5	73.70	0.62	0.66	0.81	0.64	0.26	0.79
	10	78.45	0.60	0.81	0.91	0.69	0.22	0.85
	15	84.91	0.74	0.84	0.92	0.79	0.15	0.91
	30	90.23	0.79	0.89	0.95	0.84	0.09	0.94
	60	92.02	0.70	0.95	0.99	0.81	0.08	0.98
	90	93.86	0.75	0.95	0.99	0.84	0.06	0.99

Table A.6: XGBoost Results Table 1: results of XGBoost implemented on stocks of AAPL, AMS, AMZN, FB, and MSFT.

Company Name	Trading Window	Accuracy	Recall	Precision	Specificity	F-Score	Brier Score	AUC
AAPL	3	55.99	0.80	0.56	0.29	0.66	0.44	0.57
	5	59.64	0.77	0.60	0.40	0.67	0.40	0.62
	10	61.50	0.79	0.61	0.41	0.69	0.39	0.66
	15	64.48	0.78	0.64	0.48	0.70	0.36	0.70
	30	68.82	0.85	0.68	0.47	0.76	0.31	0.76
	60	72.61	0.90	0.70	0.49	0.79	0.27	0.83
	90	77.13	0.88	0.77	0.61	0.82	0.23	0.86
AMS	3	58.99	0.78	0.59	0.36	0.67	0.41	0.61
	5	59.87	0.78	0.58	0.41	0.67	0.40	0.64
	10	62.44	0.73	0.60	0.52	0.66	0.38	0.69
	15	64.83	0.69	0.61	0.61	0.65	0.35	0.70
	30	66.81	0.65	0.68	0.69	0.66	0.33	0.74
	60	73.82	0.81	0.71	0.67	0.75	0.26	0.82
	90	77.03	0.79	0.76	0.75	0.77	0.23	0.85
AMZN	3	55.92	0.80	0.56	0.30	0.65	0.44	0.59
	5	58.10	0.82	0.57	0.32	0.67	0.42	0.63
	10	58.52	0.83	0.57	0.32	0.68	0.41	0.66
	15	62.63	0.88	0.60	0.33	0.72	0.37	0.72
	30	67.21	0.90	0.66	0.36	0.76	0.33	0.77
	60	76.22	0.86	0.77	0.62	0.81	0.24	0.84
	90	82.69	0.92	0.82	0.68	0.87	0.17	0.88
FB	3	59.31	0.78	0.60	0.37	0.68	0.41	0.63
	5	65.05	0.79	0.65	0.48	0.71	0.35	0.72
	10	68.75	0.80	0.71	0.53	0.75	0.31	0.75
	15	80.84	0.86	0.85	0.71	0.86	0.19	0.86
	30	86.88	0.93	0.87	0.75	0.90	0.13	0.93
	60	88.76	0.98	0.89	0.57	0.93	0.11	0.89
	90	94.44	0.98	0.96	0.72	0.97	0.06	0.95
MSFT	3	55.18	0.82	0.55	0.26	0.66	0.45	0.56
	5	56.55	0.86	0.56	0.22	0.68	0.43	0.60
	10	60.50	0.85	0.60	0.31	0.70	0.40	0.67
	15	64.60	0.86	0.64	0.35	0.74	0.35	0.70
	30	69.17	0.80	0.71	0.55	0.75	0.31	0.77
	60	74.04	0.83	0.75	0.61	0.79	0.26	0.83
	90	78.03	0.83	0.79	0.72	0.81	0.22	0.87

Table A.7: XGBoost Results Table 2: results of XGBoost implemented on stocks of NKE, SNE, TATA, TWTR, and TYO.

Company Name	Trading Window	Accuracy	Recall	Precision	Specificity	F-Score	Brier Score	AUC
NKE	3	56.53	0.84	0.57	0.23	0.68	0.43	0.58
	5	56.64	0.85	0.57	0.22	0.68	0.43	0.60
	10	60.01	0.92	0.59	0.21	0.72	0.40	0.67
	15	61.93	0.96	0.61	0.15	0.74	0.38	0.67
	30	68.50	0.98	0.67	0.21	0.79	0.31	0.73
	60	72.25	0.90	0.72	0.45	0.80	0.28	0.79
	90	78.90	0.94	0.77	0.55	0.84	0.21	0.87
SNE	3	53.72	0.69	0.53	0.38	0.60	0.46	0.57
	5	56.86	0.71	0.57	0.42	0.63	0.43	0.60
	10	60.16	0.74	0.59	0.46	0.66	0.40	0.66
	15	62.50	0.74	0.60	0.51	0.67	0.38	0.68
	30	66.10	0.64	0.69	0.68	0.66	0.34	0.72
	60	70.21	0.87	0.67	0.51	0.76	0.30	0.77
	90	70.50	0.84	0.67	0.56	0.74	0.30	0.80
TATA	3	55.80	0.40	0.53	0.69	0.46	0.44	0.59
	5	60.51	0.40	0.61	0.78	0.48	0.39	0.66
	10	68.56	0.44	0.76	0.88	0.56	0.31	0.74
	15	70.91	0.50	0.75	0.87	0.60	0.29	0.80
	30	74.11	0.58	0.79	0.88	0.67	0.26	0.85
	60	84.01	0.75	0.83	0.90	0.79	0.16	0.92
	90	87.22	0.81	0.86	0.91	0.83	0.13	0.94
TWTR	3	71.07	0.70	0.69	0.72	0.70	0.29	0.74
	5	73.60	0.75	0.67	0.72	0.71	0.26	0.81
	10	81.12	0.80	0.78	0.82	0.79	0.19	0.87
	15	84.02	0.86	0.78	0.82	0.82	0.16	0.91
	30	84.66	0.77	0.81	0.89	0.79	0.15	0.93
	60	85.06	0.73	0.79	0.91	0.76	0.15	0.93
	90	83.02	0.79	0.80	0.86	0.79	0.17	0.93
TYO	3	59.09	0.32	0.54	0.79	0.40	0.41	0.61
	5	66.12	0.39	0.57	0.82	0.46	0.34	0.67
	10	71.22	0.41	0.75	0.91	0.53	0.29	0.77
	15	78.38	0.54	0.83	0.93	0.65	0.22	0.86
	30	84.03	0.64	0.82	0.93	0.72	0.16	0.90
	60	88.72	0.56	0.94	0.99	0.70	0.11	0.91
	90	92.15	0.65	0.97	0.99	0.78	0.08	0.97