Journal of *Risk and Financial Management*

Article

Forecasting a Stock Trend Using Genetic Algorithm and Random Forest

Rebecca Abraham [1,*], Mahmoud El Samad [2], Amer M. Bakhach [2], Hani El-Chaarani [3], Ahmad Sardouk [4], Sam El Nemar [5] and Dalia Jaber [2]

[1] Huizenga College of Business, Nova Southeastern University-SBE, 3301 College Avenue, Fort Lauderdale, FL 33319, USA
[2] School of Arts and Sciences, Lebanese International University, Mouseitbah, Mazara P.O. Box 146404, Lebanon; mahmoud.samad@liu.edu.lb (M.E.S.); amer.bakkach@liu.edu.lb (A.M.B.); dalia.jaber@liu.edu.lb (D.J.)
[3] College of Business Administration, Tripoli Campus, Beirut Arab University, Beirut P.O. Box 11-50-20, Lebanon; h.shaarani@bau.edu.lb
[4] Faculty of Economics and Business Administration, Tripoli Campus, Lebanese University (UL), Beirut P.O. Box 6573/14, Lebanon; ahmad.sardouk@gmail.com
[5] Faculty of Business Administration, AZM University, Tripoli P.O. Box 1010, Lebanon; snemer@azmuniversity.edu.lb
* Correspondence: abraham@nova.edu

Abstract: This paper addresses the problem of forecasting daily stock trends. The key consideration is to predict whether a given stock will close on uptrend tomorrow with reference to today's closing price. We propose a forecasting model that comprises a features selection model, based on the Genetic Algorithm (GA), and Random Forest (RF) classifier. In our study, we consider four international stock indices that follow the concept of distributed lag analysis. We adopted a genetic algorithm approach to select a set of helpful features among these lags' indices. Subsequently, we employed the Random Forest classifier, to unveil hidden relationships between stock indices and a particular stock's trend. We tested our model by using it to predict the trends of 15 stocks. Experiments showed that our forecasting model had 80% accuracy, significantly outperforming the dummy forecast. The S&P 500 was the most useful stock index, whereas the CAC40 was the least useful in the prediction of daily stock trends. This study provides evidence of the usefulness of employing international stock indices to predict stock trends.

Keywords: computational or mathematical finance; stock trend prediction; random forest; genetic algorithm; features selection

check for updates

Citation: Abraham, Rebecca, Mahmoud El Samad, Amer M. Bakhach, Hani El-Chaarani, Ahmad Sardouk, Sam El Nemar, and Dalia Jaber. 2022. Forecasting a Stock Trend Using Genetic Algorithm and Random Forest. *Journal of Risk and Financial Management* 15: 188. https://doi.org/10.3390/jrfm15050188

Academic Editor: Thanasis Stengos

Received: 22 February 2022
Accepted: 14 April 2022
Published: 19 April 2022

1. Introduction

Stock market movements are influenced by many exogenous variables, such as political and geopolitical events, exchange rates, movements of other stock markets, the economic environment, firm policies, and the psychology of investors (Gidofalvi 2001; Nobel Prize Committee 2013; El-Chaarani 2016; El-Chaarani 2019).

For efficient market theory (Fama 1965; Fama 1970; Fama 1998), all relevant information must be reflected in efficient stock markets. In the weak-form market efficiency, prices of stocks reflect all information of past prices. In semi-strong market efficiency form, prices of stocks reflect all of the available public information. We employ uptrend forecasting, which contrasts with market efficiency, as it includes real time information that does not mirror the theoretical conditions.

Dynamic markets with nonlinear stock price movements, along with the multiplicity of predictors makes forecasting a stock's trend challenging. In fact, an efficient forecasting solution in the stock market can play a crucial role in motivating people toward stock trading (Sharma et al. 2017).

Artificial Neural Networks (ANN) and the Support Vector Machine (SVM) are the most commonly used machine learning algorithms for forecasting stocks (Guresen et al. 2011; Hoseinzade 2019; Kara et al. 2011; Wang and Wang 2015). Many Artificial Neural Networks tend to predict the next day's closing price (Li and Liao 2017). Certain Artificial Neural Networks-based models have been enriched with other algorithms to boost the accuracy of the forecast (Nair et al. 2011). Likewise, Support Vector Machine models have been developed to forecast stock trends (Reddy and Sai 2018). Some proposals combined the Support Vector Machine, or Artificial Neural Networks with preprocessing techniques, following meta-heuristics algorithms to find the optimal machine learning parameters, the Artificial Neural Networks architecture, and a set of input features (Asadi et al. 2012; Sedighi et al. 2019; Zhang and Wu 2009). Recently, deep learning techniques such as Convolutional Neural Network (CNN), Deep Multilayer Perceptron (MLP), Deep Belief Network (DBN), Recurrent neural network (RNN), Long Short-Term Memory (LSTM), and Generative Adversarial Network (GAN) have proved to be applicable to stock studies (Aloud 2020; Sang and DiPierro 2019; Selvin et al. 2017; Zhanga et al. 2019). The features selection is a key factor for the knowledge discovery process. Its importance is derived from its role in improving the accuracy and efficiency of the prediction model by selecting the relevant variables and reducing the dimensionality of the datasets (Mao et al. 2016; Sugunnasil and Somhom 2010). Yet, features selection models suffer from the drawback of using isolated features to forecast trends. There is a gap in the literature for stock trend models that not only use features selection, but include international stock indices to forecast trends, as such models capture the essence of worldwide market movements, rather than isolated features. This paper offers such a model.

Our model is as follows:

1. Select a set of international stock indices. In this work, the focus is on the NIKKEI 225, CAC 40, DAX, and S&P 500;
2. Select one stock to be forecasted, such as Apple;
3. Employ a Genetic Algorithm for feature selection. The objective here is to decide which historical prices are significant in forecasting the stock's trend;
4. Finally, we consider the Random Forest (RF) algorithm to find hidden relationships between the selected features (from Step 3), and the stock's trend.

This remainder of this paper is organized as follows. Section 2 is a review of literature. Section 3 describes our approach for forecasting a stock's trend. Section 4 discusses materials and methods. Section 5 presents the results, while the conclusions are described in Section 6.

2. Review of Selected Literature

In this section, we address several studies that have attempted to develop machine learning models to predict stock prices. Artificial Neural Network prediction models have been proposed in many research works, such as Chandan et al. (2016). However, the noisy behavior of stock markets constructs an obstacle for the Artificial Neural Networks, leading to convergence to suboptimal solutions (Hoseinzade 2019). To solve this problem, Kara et al. (2011) suggested that a Support Vendor Machine preprocessing model can help in eliminating irrelevant features. With respect to predicting stock trends, Reddy and Sai (2018) proposed a Support Vendor Machine and Radial Basis Function approach to forecast stock prices in large as well as in small capitalizations. Their predictor is trained based on the available historical data to predict the next day's data. While the obtained numerical results showed high efficiency of the algorithm, its drawback is that the solution assumes four fixed features without any specific engineering or optimization. The experimentation relies on online data without addressing its quality.

Hoseinzade (2019) suggested a model, called CNNpred, that can be applied to a collection of data from different sources, from different markets. The approach uses feature extraction to predict the next day's trend of movement for specific indices, including the S&P 500, NASDAQ, DJI, NYSE, and the RUSSELL 3000. Similarly, Chen (2018) used a conv1D function to process 1D data in the convolutional layer. To improve the results, the proposed model used preprocessed stock data as input. The work goal was to forecast stock prices in the Chinese stock market. The proposed model was limited to four features as open, close, high, and low prices. The chief limitation was that the validation relied on a limited dataset.

Karathanasopoulos et al. (2019) proposed several optimization techniques to find the optimal Neural Network hierarchy to forecast 12 Exchange Traded Funds (ETFs). They considered three optimization approaches, namely genetic algorithm, differential evolution, and the particle swarm optimizer. They also considered three multilayer perceptron, recurrent neural networks, and radial basis function neural networks. Their results showed that differential evolution was the optimal method, with the highest forecasting accuracy.

The study most analogous to this paper is the one by Jiao and Jakubowicz (2017). This study predicted the daily direction of stock price movement. The authors considered predicting stock movement as a binary classification problem. They studied 463 stocks, constituents of the S&P 500, and 8 international indices. International indices included three Asian indices (Nikkei 225, Hang Seng, and All Ords), two European indices (DAX, FTSE 100), and three US indices (NYSE Composite, Dow Jones Industrial Average, and S&P500). Thus, when the daily return was positive (greater than zero), the mean price direction was uptrend. When the daily return was negative (less than zero), the mean price direction was downtrend. After that, they used a lag operator to extract more features from stock indices, in addition to more than 200 technical indicators, as input features into a classifier. Then, they employed genetic algorithm-based feature selection, to use the selected features as input into a classifier. The authors used four classification algorithms to compare their prediction performance. The classification algorithms used included Random Forest, Gradient Boosted Trees, Artificial Neural Networks, and logistic regression. However, Jiao and Jakubowicz (2017) did not examine the usefulness of international stock indices to forecast stock trends, which we relied on in our study. Additionally, we proposed a genetic algorithm-based approach to select only helpful features. We extracted 10 lag features from each international stock index, beside the stock historical data of the stock whose trend we wished to predict. Furthermore, they classified trends of stock into uptrend if the daily return price change was positive (greater than zero), and downtrend, if the daily return price change was negative (less than zero). In our study, we classified trends of stock as uptrend if the daily return price change is greater than five tenths percentage (0.5%). If the daily return price change was less than five tenths percentage (0.5%), the classification was Not-Uptrend. Our objective was to ensure that the trader would not sustain a loss. Our approach answered the question of whether international stock indices could be useful in forecasting stock trends, whereas the Jiao and Jakubowicz (2017) approach could not provide such insight.

Sable et al. (2017) provided a short-term prediction model, using the Genetic Algorithm, and evolutionary strategies, predicting the price of eight scripts, with six attributes for each script (Opening Price, Closing Price, Highest Price, Lowest Price, Volume, and Adjusted Closing Price). The eight scripts reflect US-based companies. This paper does not address other international markets. Neither does it mention the specific reason behind the chosen six attributes.

Shen and Shafiq (2020) proposed a prediction model for the stock market price trend. The proposal relies on a customization of feature engineering and deep learning. The pillars of the proposal are various techniques of feature engineering with a fine-tuned system, instead of just a deep learning model. The work assessment relies on only the Chinese stock market. While the paper has high scientific value, applying the same approach to other international markets is still to be assessed.

Wanjawa and Muchemi (2014) proved that the configuration model 5:21:21:1 can achieve very good prediction accuracy. The assessment was done on 1000 records trained on 130 K cycles. The training percentage was 80%. Using Artificial Neural Networks (ANN) to predict three stocks on the New York Stock Exchange (NYSE), with Encog and Neuroph for validation, the prediction was achieved with error range [0.71%–2.77%].

Soni et al. (2022), Rouf et al. (2021), Rahul et al. (2020), and Tawarish and Satyanarayana (2019) provide interesting surveys on the ML techniques used for stock market prediction. Almost all of these surveys show that a large percentage of proposals use Support Vector Machine (SVM), fewer use Genetic Algorithm, and fewer use the Random Forest. While these surveys support the novelty of our approach, they do not discuss any work that combines Genetic Algorithm and Random Forest, or international stock indices. Wang et al. (2021) did another survey, which focuses on work done using Fuzzy Cognitive Mapping, and the Deep Neural Network. This review is limited to research work with datasets either spanning several years, or those measured over short-term periods.

3. Our Approach for Forecasting a Stock's Trend

3.1. Problem Formulation

This research focused on forecasting whether a stock would exhibit an uptrend tomorrow, with reference to today's stock price. More formally, r_{t+1} denoted tomorrow's price change relative to today's closing price. This value of r_{t+1} is also indicated by ΔP_{t+1} as expressed in Equation (1).

$$r_{t+1} = \Delta P_{t+1} = \frac{P_{t+1} - P_t}{P_t}, \tag{1}$$

where, P_t = today's closing price.

P_{t+1} = tomorrow's closing price.

$$Trend_{t+1} = \begin{cases} Uptrend & if\ r_{t+1} \geq 0.5\% \\ Not-Uptrend & if\ r_{t+1} < 0.5\% \end{cases} . \tag{2}$$

Trend$_{t+1}$ could be either *Uptrend* or *Not-Uptrend*. Tomorrow's trend could be considered to be an '*Uptrend*' if the relative price change, r_{t+1}, exceeds a cut-off threshold of 0.5%. The cut-off threshold of 0.5% is employed to ensure that trading of the stock could provide positive returns after deducting transaction costs[1]. If the relative price change, r_{t+1} is less than the anticipated transaction costs, then the trader could easily have a loss, even if the forecast is accurate.

3.2. The Forecasting Model

The objective of this research model was to forecast the variable *Trend*$_{t+1}$.

3.2.1. Step One: Feature Engineering

The Feature Engineering step consisted of collecting and composing a set of features that would be considered in forecasting a stock's trend. Four international stock indices are considered as the main features, including the S&P 500 (United States), NIKKEI 225 (Japan), CAC40 (France), and DAX (Germany). These indices are chosen for their prominence in the global stock markets, with the S&P 500, the index of 500 US companies with highest market capitalization, NIKKEI 225, the index of the leading 225 Japanese companies, the CAC40, with 40 of the stocks with largest market capitalization in France, and DAX for 30 major German securities.

'Lag operation' is employed to extract features from stock indices that contained price change data from prior time periods. Shifting the time series historical data of each stock index to prior periods constituted the 'lag operation'. For example, in Equation (3), historical price changes of the S&P 500 are shifted to one prior period.

$$\Delta \text{S\&P 500}_t^1 = 100 \times \frac{\text{S\&P500}_t - \text{S\&P500}_{t-1}}{\text{S\&P500}_{t-1}}, \tag{3}$$

where:

$\text{S\&P500}_t$ = closing price of S&P500,

$S\&P500_{t-1}$ = yesterday's closing price,
$\Delta S\&P500_t^1$ = today's price change.

The 'lag operation' is employed to the time series of price changes for each stock index 10 times to get 10 lag features. More specifically, historical data of price changes of each stock index (such as the S&P 500) are shifted 10 times to get 10 lag features. To compute the 10 return lags of the S&P 500, 10 values for parameter *i* (1, 2, ..., 10) are considered for the S&P 500 index. For instance, if *i* = 4 was set, then the price change of S&P500 four days ago is referred to, with reference to today's price. In other words, the closing price four days before today's closing price ($S\&P500_{t-4}$).

$$\Delta S\&P500_t^i = 100 \times \frac{S\&P500_t - S\&P500_{t-i}}{S\&P500_{t-i}}. \quad (4)$$

The process of shifting the historical data of price changes for international stock indices is known as 'distributed lags analysis'.

Table 1 illustrates the 10 lag features extracted from each stock index. For each international stock index, historical data of its price changes over the past 10 days are considered. For example, the value of 'CAC40_1' can be denoted as $\Delta CAC40_t^1$ and computed based on Equation (4). The first row of Table 1 refers to the 10 features corresponding to the 10 lags of the stock market 'CAC40.' For example, 'CAC40_2' in the first row, indicates the second feature corresponding to lag 2; that is, the price changes of CAC40 stock index two days ago. The value of CAC40_2 can be computed by replacing *i* with value of two. The second row denotes 10 features corresponding to the 10 lags of 'DAX'. The same interpretation holds true regarding the third row, the US stock index, 'S&P500,' and the Japanese stock index, and 'NIKKEI255,' in the fourth row.

Table 1. The 10 lag features extracted from each global stock index in the past 10 days.

	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
CAC40	CAC 40_1	CAC 40_2	CAC 40_3	CAC 40_4	CAC 40_5	CAC 40_6	CAC 40_7	CAC 40_8	CAC 40_9	CAC 40_10
DAX	DAX_1	DAX_2	DAX_3	DAX_4	DAX_5	DAX_6	DAX_7	DAX_8	DAX_9	DAX_10
S&P500	S&P 500_1	S&P 500_2	S&P 500_3	S&P 500_4	S&P 500_5	S&P 500_6	S&P 500_7	S&P 500_8	S&P 500_9	S&P 500_10
NIKKEI225	NIKKEI 225_1	NIKKEI 225_2	NIKKEI 225_3	NIKKEI 225_4	NIKKEI 225_5	NIKKEI 225_6	NIKKEI 225_7	NIKKEI 225_8	NIKKEI 225_9	NIKKEI 225_10

Each feature consists of price changes of the stock index at the prior day corresponding to the lag number.

In addition, the historical data of the stock whose trend we predicted is also considered. The daily price changes of the stock, using historical data, is computed, as in Equation (5). More precisely, the lag operation to shift the historical data of the price changes several steps back is employed. Ten lag features shifted from the historical data of price changes of the considered stock are extracted. Therefore, the 10 values of *i* (1, 2, . . . ,10) are considered. Each lag feature represents price changes of the stock on the previous day, corresponding to the lag number. As an example, LAG 2, of the considered stock (such as AAPL) represents price changes of stock AAPL two days ago. LAG 2's value, denoted as $\Delta AAPL_t^2$, is obtained by setting the value of *i* to 2. Shifting the time series of a stock's historical data is known as autocorrelation (Salkind 2010). It computes the correlation of a time series with the same series at prior time steps. It tests if the current value of the stock is affected by shifted values of the time series of the stock itself at prior time steps.

$$\Delta AAPL_t^i = 100 \times \frac{AAPL_t - AAPL_{t-\mathrm{i}}}{AAPL_{t-\mathrm{i}}}. \quad (5)$$

Ten lag features extracted from the historical data of the price change of AAPL stock are presented in Table 2. The first column represents the name of the stock, whose trend is to be forecasted. The second column (LAG1) represents the lag index of the historical data of the price change of the AAPL stock the previous day. The third column (LAG2) represents the historical price change of the AAPL stock, shifted two days prior to the present. In fact, the ΔAAPL_t^2 can be computed by replacing i with 2 in Equation (5). The idea is to keep shifting the historical data of price changes of the AAPL stock till it reaches 10 days prior to the present day.

Table 2. The 10 lag features extracted from AAPL stock in the past 10 days.

Stock Name	**LAG1**	**LAG2**	**LAG3**	**LAG4**	**LAG5**	**LAG6**	**LAG7**	**LAG8**	**LAG9**	**LAG10**
AAPL	AAPL_1	AAPL_2	AAPL_3	AAPL_4	AAPL_5	AAPL_6	AAPL_7	AAPL_8	AAPL_9	AAPL_10

The price changes of AAPL stock are denoted by each lag feature on the prior day corresponding to the lag number.

3.2.2. Step Two: Feature Selection Using Genetic Algorithm

The Feature Engineering step in the previous section resulted in a total of 50 features. These 50 features are composed of 10 lag features extracted from the historical data of each international stock index in addition to the 10 lags associated with the considered stock itself. In this section, this study identifies which of these 50 features are truly useful in forecasting the stock's trend, $Trend_{t+1}$.

See Table 3 for a one sample chromosome represented as zeroes and ones. Create an initial population. The initial population is created randomly by the Genetic Algorithm. A population is a set of individuals, where an individual is composed of a subset of features. An individual, also known as a chromosome, contains a number of genes, where a gene is a feature in our problem. The number of genes is equal to the number of input features (that is 50, in our case). Subsequently, binary coding is applied for the genes with a gene value of zero or one. A gene value of 1 indicates that the corresponding feature is labeled as useful in forecasting the considered stock, otherwise it will not be employed for the forecast.

Table 3. One sample chromosome of population represented as a set of zeroes and ones.

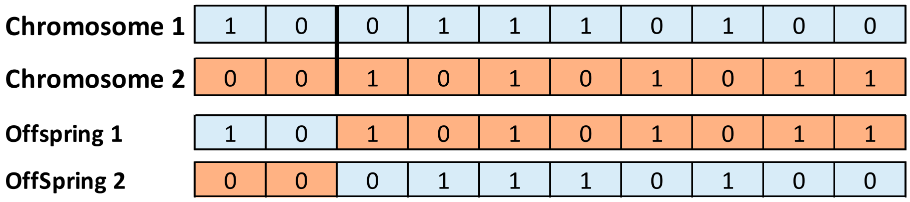
Name of Index or Stock	**LAG1**	**LAG2**	**LAG3**	**LAG4**	**LAG5**	**LAG6**	**LAG7**	**LAG8**	**LAG9**	**LAG10**
CAC 40	1	1	1	0	0	1	1	0	0	0
DAX	0	0	1	0	0	1	0	0	0	1
S&P500	0	0	1	0	1	0	1	1	1	0
NIKKEI225	1	0	0	1	0	0	1	1	0	1
Stock (AAPL)	1	1	0	1	0	0	0	1	1	1

All lags with value one are selected for the forecast. For instance, LAG3, LAG6, and LAG10 columns, associated with the stock index DAX (shown in bold in the first column), are labeled as useful, and are thus employed to conduct the forecast. Other LAG features of DAX stock are eliminated in this chromosome. The objective is to find the best set of features (distribution of ones and zeroes) that can forecast $Trend_{t+1}$.

Calculate the objective function. The ability of each chromosome in the population to predict $Trend_{t+1}$ is evaluated by the objective function. The relevance of each chromosome, and its ability to predict $Trend_{t+1}$ is assessed. For this purpose, the Random Forest algorithm is employed. The Random Forest is a well-known machine learning algorithm, which can be used for classification and regression purposes. The Random Forest is employed to find out the relation between the features represented in each chromosome and $Trend_{t+1}$.

Copy the best 10% of the chromosomes. The top 10% of the chromosomes with the highest accuracy over the validation dataset, from the previous step, are copied to the next new population.

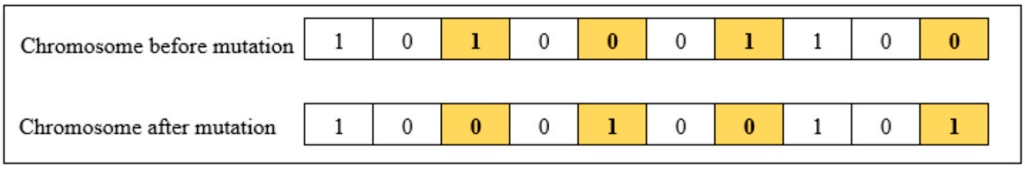
Crossover. See Figure 1. Once the accuracy of all the chromosomes is evaluated, they are subjected to a crossover process. The crossover process is used to reproduce new chromosomes (distribution of ones and zeroes), from old chromosomes. In this process, two chromosomes, (known as the parents), are selected to be mixed, to produce two new chromosomes (known as the offspring). To select the parent chromosomes, the roulette wheel selection algorithm is used. The roulette wheel algorithm is applied twice to get the chromosomes of two parents. After that, a single point crossover is applied. The crossover point is selected randomly. This point determined the position at which each chromosome is divided. Each chromosome is divided into two parts according to a random crossover point. Then, the crossover process merged the first part of the first parent with the second part of the second parent, followed by merging the first part of the second parent with the second part of the first parent, resulting in new chromosomes (offspring), produced with new features.



Chromosome 1	1	0	0	1	1	1	0	1	0	0
Chromosome 2	0	0	1	0	1	0	1	0	1	1
Offspring 1	1	0	1	0	1	0	1	0	1	1
OffSpring 2	0	0	0	1	1	1	0	1	0	0

Figure 1. A sample single-point crossover, with the point's index set randomly to two.

Mutation. See Figure 2. Following the new generation of crossover chromosomes, random changes in the selected features of one chromosome are made by the mutation process. The objective of mutation is to avoid a local optimum in the search space. The mutation process flipped the value of some genes' bits, for a random number of chromosomes that have been reproduced using the aforementioned crossover process.



Chromosome before mutation	1	0	**1**	0	**0**	0	**1**	1	0	**0**
Chromosome after mutation	1	0	**0**	0	**1**	0	**0**	1	0	**1**

Figure 2. Mutation example with 4 genes to be mutated (genes at index 3, 5, 7, and 10).

New Population. After the crossover process, and after the mutation process are completed, the new population is ready for the next iteration of the Genetic Algorithm. This new population is composed of:

- The top 10% of the chromosomes copied from the initial population that provided the highest accuracies;
- The 40% of the chromosomes selected from the new offspring generated by the crossover process, subjected to the mutation process;
- The 50% of the chromosomes selected from the new offspring generated by the crossover process, without subjection to the mutation process. These fifty percent are selected using the roulette wheel algorithm, with each offspring having selection chances as per its accuracy.

The Genetic Algorithm STOP. The optimization cycle is repeated, until the accuracy of forecasting over the validation dataset did not increase by more than 0.5% over 10 successive iterations. A single chromosome to be employed to forecast the stock's trend $Trend_{t+1}$ is returned by the Genetic Algorithm feature.

3.2.3. Step Three: Conducting Forecasts

The Random Forest algorithm is applied to predict $Trend_{t+1}$. Random Forest creates a forest with large number of decision trees for classification in training. Each node in a decision tree represents a feature. The root node of the tree consists of the best feature that splits the training samples. The decision node (or internal node) denotes a test on a feature. More particularly, it asks yes or no questions to test the feature and split the points for the decision into sub-nodes. The branches denote the decision rule represented as 'yes or no′ branches (outcome of the test on the training dataset). Each leaf node (terminal node) denotes the categorical, up or down, classification decision. Leaf nodes do not split, as they are the last structure in the tree. Figure 3 represents the flowchart of a decision tree.

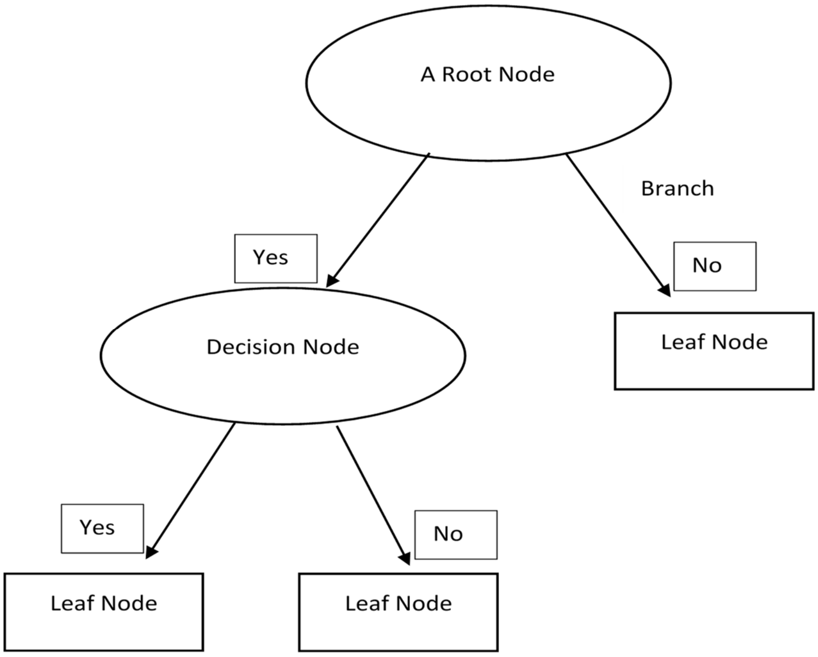


Figure 3. A sample random forest decision tree structure.

Each tree is built on a random selected sample and a subset of features from a training dataset. Each node in a tree splits the path into a ‘yes’ branch or ‘no’ branch. The tree started by placing the significant feature, best in splitting the samples in the training set, as a root node. The tree evaluated the importance of the feature by employing a selection algorithm, such as the Gini Index, or Information Gain. Each node in the decision tree was split into sub-nodes. The root node divided the training set samples into homogenous subsets by a ‘yes or no’ feature testing. It compared root feature samples to the records values of next node. Next, the sample set that met the criteria, followed the ‘yes’ branch, while the rest followed the ‘no’ branch. Comparison between node feature values and next node records continued until we reached leaf nodes and had a classification output. Random Forest collected all the class labels (decision outputs) from all decision trees, and selected the final decision based on majority votes as the best classification result.

The features of one sample chromosome denoted as zeroes and ones, in which this chromosome is selected randomly, is illustrated in Table 4. The CAC40 row indicates the employed lag features of CAC40, denoted as ‘1’, to forecast $Trend_{t+1}$. The same interpretation applies to the remaining rows. Next, all selected features, denoted as one from the selected chromosomes, are sent to the Random Forest classifier to build decision trees. Each tree had a categorical classification decision (Uptrend or Not-Uptrend). Random Forest selected the final classification decision, according to the majority votes of the trees. This process ran recursively on each chromosome, improving the accuracy of predictions with each iteration. Finally, the resulting decision tree model is employed to an out-of-sample dataset to evaluate our model in terms of accuracy of performance.

Table 4. The features of a one sample chromosome, represented as zeroes and ones.

Stock Name	LAG1	LAG2	LAG3	LAG4	LAG5	LAG6	LAG7	LAG8	LAG9	LAG10
CAC 40	0	1	0	0	0	1	0	0	0	1
DAX	0	0	1	0	0	0	1	0	0	1
S&P500	0	0	1	1	1	0	1	0	1	0
NIKKEI225	1	0	0	1	0	1	1	0	0	1
Stock (APPL)	0	0	0	1	0	0	0	0	1	0

In sum, our approach comprised the following steps:

1. The basic features extracted from historical data of four international stock indices (following the distributed lag analysis concept), and the historical data of the stock itself (following the concept of autocorrelation), are calculated;
2. A Genetic Algorithm Approach is followed to identify the most useful features to predict a stock's trend, using a training or validation dataset;
3. A Random Forest classifier is trained, using training and validation datasets (Figure 4), to find the relationships leading to a stock's trend. Then, the stock trend is forecasted over the out-of-sample testing dataset using the produced Random Forest decision tree.

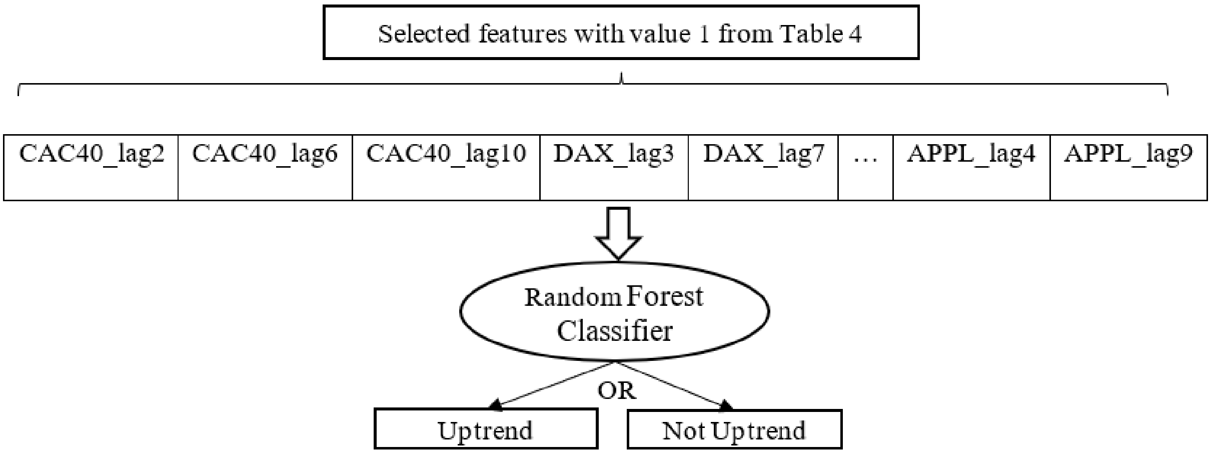


Figure 4. Random forest classifier.

4. Materials and Methods

4.1. Data Selection

See Table 5. Fifteen stocks listed are considered, belonging to the Technology, Finance, and Healthcare industries. The daily stock closing prices for each stock are listed, sampled from 2 January 2018 to 30 June 2019.

Table 5. The distribution of the 15 selected stocks across 3 sectors.

Sector	Selected Stocks				
Technology	FB	APPL	BABA	JD	EVTC
Finance	ESNT	GNW	GTY	EVR	AVAL
Healthcare	HCA	WAT	ALC	ABBV	ABC

It is important to evaluate stock forecasts and trades using a set of assets with different trends (Prado 2011). The performance of the research forecasting model under different market scenarios is tested by such variation in the selected dataset, thereby avoiding bias towards particular patterns. The descriptive statistics of the daily price changes for all considered stocks are reported in Table 6. It should be noted that some stocks have positive means (such as, GNW, EVTC, and ALC), whereas the daily returns of other stocks have negative means (such as, EVR, AVAL, and ABC). Additionally, positive skewness is reported among certain stocks (such as, GNW, AVAL, ALC), whereas others have negative skewness (such as, ESNT, FB, ABBV). The considered stocks exhibited different trends during the period, 2 January 2018 to 30 June 2019.

Table 6. Descriptive statistics of the daily rate of returns r_{t+1} of the selected stocks.

Sector	Stock Symbol	Mean	Standard Deviation	Skewness	Kurtosis
Finance	ESNT	0.000241479	0.020302896	−1.72247705	11.43985993
	GNW	0.001079841	0.032141733	2.393306874	16.36510994
	GTY	0.000569729	0.013049528	−0.266608106	2.501580582
	EVR	−0.00019415	0.019682099	0.307557925	2.626563921
	AVAL	−0.000267656	0.016610293	0.812591073	6.70001155
Technology	FB	0.000302066	0.023904315	−1.335291055	14.05632121
	APPL	0.000689175	0.018406884	0.129020529	3.017014388
	BABA	−0.00039628	0.022833273	0.017442468	0.446496642
	JD	−0.001297827	0.027757257	−0.153793418	0.80117389
	EVTC	0.002466215	0.021055482	1.298376309	10.40076212
Healthcare	HCA	0.000918586	0.016395269	−0.28991327	8.3983388
	WAT	−0.0000783316	0.017568363	−0.240412285	15.40554447
	ALC	0.075033114	0.438104882	5.904849568	34.22098852
	ABBV	−0.001094373	0.02099967	−2.117082735	14.22452989
	ABC	−0.000296482	0.01893862	−0.213177629	3.212242095

Furthermore, a dataset of four international stock indices (S&P500, NIKKEI 225, CAC 40, DAX) over the period from 2 January 2018 to 30 June 2019 is selected and prepared. Distributed lag analysis and autocorrelation to extract 10 lag features from each considered stock and each stock index, are employed. The resulting 50 lag features are employed for training and testing our model to forecast $Trend_{t+1}$.

4.2. Model Training and Testing Process

A rolling window approach has been proposed to improve the reliability of the trading strategy (Prado 2011) The rolling window approach is usually used for evaluating trading systems. Data is divided into overlapping training sets. In each cycle, each is moved forward through the time series. Stricter models ensue due to more frequent retraining, and large out-of-sample data sets (increasing training processing requirements, but with models that adapt more quickly to the changing market conditions). For this study, training and testing of the model is conducted using a monthly rolling window (see Figure 5).

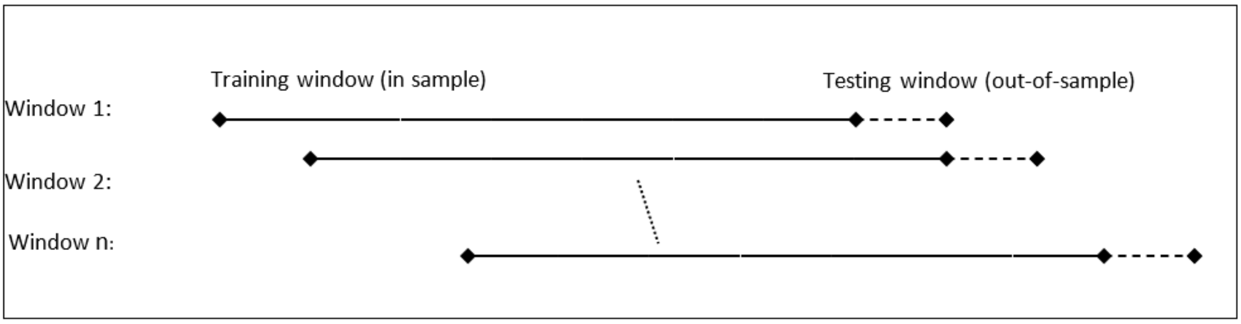


Figure 5. Illustration of 12 rolling windows for the training, validation, and testing sets.

See Table 7. The dataset in Section 4.1 is sampled on a daily basis over 17 months, between 2 January 2018 and 30 June 2019. The dataset is divided into multiple overlapping training sets. Each set is composed of two sub-sets. The time length of the first subset was five months, which is subsequently used for model training. The second set is a one-month time window for validation. In fact, prediction is possible, on a daily or weekly basis, but will require high levels of computation for testing, which are beyond the scope of this paper. By adopting monthly testing, sufficient variations in stock movements are being included, as such variations are generated by a dynamic stock market. In future work, daily or weekly testing will be employed to capture a larger number of stock prices. Excessive computation would also be required for shorter training sets, which, although challenging at present, will be considered in future research.

Table 7. Dataset splitting into a rolling window from 2 January 2018 to 30 June 2019.

Rolling Windows ID	Training (5 Months)	Testing (1 Month)
1	From 1/2/2018 to 30/6/2018	From 1/7/2018 to 31/7/2018
2	From 1/3/2018 to 31/7/2018	From 1/8/2018 to 31/8/2018
3	From 1/4/2018 to 31/8/2018	From 1/9/2018 to 30/9/2018
4	From 1/5/2018 to 30/9/2018	From 1/10/2018 to 31/10/2018
5	From 1/6/2018 to 31/10/2018	From 1/11/2018 to 30/11/2018
6	From 1/7/2018 to 30/11/2018	From 1/12/2018 to 31/12/2018
7	From 1/8/2018 to 31/12/2018	From 1/1/2019 to 31/1/2019
8	From 1/9/2018 to 31/1/2019	From 1/2/2019 to 28/2/2019
9	From 1/10/2018 to 28/2/2019	From 1/3/2019 to 31/3/2019
10	From 1/11/2018 to 31/3/2019	From 1/4/2019 to 30/4/2019
11	From 1/12/2018 to 30/4/2019	From 1/5/2019 to 31/5/2019
12	From 1/1/2019 to 31/5/2019	From 1/6/2019 to 30/6/2019

The training and validation dataset are employed to find the best features set, using the Genetic Algorithm. Table 8 lists the settings of the Genetic Algorithm module as employed in the experiments. The second subset is a one-month time window for testing the out-of-sample, dataset. Each set is moved forward one month in each round of this rolling window. 12 rolling window sets are composed.

Table 8. The genetic algorithm settings for feature selection.

Setting	Value
Size of population	500
Crossover type	Single point
Crossover rate	100%
Mutation rate	40%
Number of genes subjected for mutation within each chromosome	7
Stop criteria	10 successive iterations with less than 0.5% increasing in accuracy
Objective function	Accuracy
Number of genes in one chromosome	50

Three types of experiments to test the model are employed:

A. Evaluation of the accuracy of the model;
B. Evaluation of the usefulness of international stock indices;
C. Sensitivity analysis.

Next, the details of all of the experiments are discussed.

A. Evaluation of The Accuracy of Our Model

The objective of this experiment is to evaluate the performance of the model. Accuracy denotes the fraction of the correct classifications we obtained from our model, as measured by Equation (6). The accuracy of forecasting $Trend_{t+1}$ of each considered stock across all three sectors is measured. Twelve sets of rolling windows for each stock are trained and tested. In other words, for each stock, 12 datasets, with each dataset being split into training, are composed, with the testing of sub-datasets. The training phase consisted of feature selection. The testing sub-dataset is employed to measure the accuracy of prediction of $Trend_{t+1}$.

$$Accuracy = \frac{Number\ of\ correct\ predictions}{Total\ number\ of\ predictions}. \quad (6)$$

Furthermore, these accuracies are compared with the accuracies obtained by a dummy forecast model. A baseline measurement of performance using simple rules and different strategies to predict are given by the dummy forecast model. The dummy forecast is based on the imbalance in the dependent variable, $Trend_{t+1}$. For instance, suppose that 65% of the instances of $Trend_{t+1}$ are *Uptrend* and 35% are *Not-Uptrend*. In such a case, the accuracy of a dummy forecast is 65%. Furthermore, to validate this comparison, the nonparametric Wilcoxon signed rank test is employed. The Wilcoxon test is used to compare two matched samples with unknown distributions, where the median difference between the pairs of observations of this forecasting model and the dummy forecast are zero. The possible rejection of the null hypothesis is determined by the result of this comparison (Wilcoxon 1945).

B. Evaluation of the Usefulness of International Stock Indices

The usefulness of each international stock index to predict a stock trend is highlighted. A total of 40 lag features, based on international stock indices, are extracted. In addition, the 10 lags related to the considered stock itself. Then, those 50 features are fed into the model to start the training and testing process. The optimal chromosome with the highest accuracy was returned at the end of the training and feature engineering process. In this experiment, the optimal chromosome to determine the most and least effective stock indices for the prediction process is used. To clarify this method, the best chromosome selected by the model over one sample rolling window for one particular stock is considered in Table 9. A value of one, the selected lag features extracted from each stock index and the stock itself, are presented in Table 9. Selected lag features that correspond to the CAC40 stock index are represented in the LAG 2, LAG 6, LAG 8, and LAG 10 columns, in row CAC40. In the case of row DAX, the selected lag features are denoted in columns of LAG 3, LAG 7, and LAG 10.

Table 9. A sample of one chromosome returned by the genetic algorithm feature to forecast AAPL for a single rolling window.

Stock Name	LAG 1	LAG 2	LAG 3	LAG 4	LAG 5	LAG 6	LAG 7	LAG 8	LAG 9	LAG 10
CAC 40	0	1	0	0	0	1	0	1	0	1
DAX	0	0	1	0	0	0	1	0	0	1
S&P500	1	0	1	1	1	0	1	1	1	0
NIKKEI225	1	0	0	1	0	1	0	0	0	1
Stock (AAPL)	0	1	0	1	0	1	0	1	1	0

After obtaining the best chromosome, the number of selected lag features from each stock index and the stock itself is computed. The most and least effective stock index is determined by the number of selected features. As shown in Table 10, four lag features are selected corresponding to the CAC40 stock market. Three lag features are extracted from the DAX stock market. The highest number of selected features of seven, from all stock indices, are from the S & P 500 index, suggesting that the S&P 500 had the strongest impact on the forecasting model. Conversely, the least frequently selected lag features from all stock indices are in the DAX row.

Table 10. Number of selected lag features from each stock index according to the chromosome selected from Table 9.

Stocks	Selected Lags with Value 1
CAC 40	4
DAX	3
S&P500	7
NIKKEI225	4
Stock (APPL)	5

C. Sensitivity Analysis

In the study problem formulation, it is considered a cut-off threshold for the daily price change, r_{t+1}, of 0.5%, to consider $Trend_{t+1}$ as an Uptrend. In this experiment, a sensitivity analysis to validate the accuracy of our model's threshold is conducted. The question is "How would a different value for the cut-off thresholds affect the accuracy of our forecasting model?" For this purpose, Equation (8), as a generalized form of Equation (7), is introduced.

$$Trend_{t+1}\begin{cases} Uptrend & if\ r_{t+1} \geq 0.5\% \\ Not - Uptrend & r_{t+1} < 0.5\% \end{cases}, \tag{7}$$

$$Trend_{t+1}\begin{cases} Uptrend & if\ r_{t+1} \geq 0.5\% \\ Not - Uptrend & r_{t+1} < 0.5\% \end{cases}. \tag{8}$$

The impact on the accuracy of the model to different values of thresholds (α in Equation (8)) is analyzed. The forecasting approach is applied to the identical 15 selected stocks using multiple values of $\alpha = \{0.5, \ldots, 2.4\}$ where α rises from 0.5 to 2.4, in steps of 0.1. For each value of α, the overall average accuracy of our model for all stocks over the 12 rolling windows following the steps of Section 4.2 is computed.

5. Results

In this section we will describe and discuss the results of the three types of experiments, A, B, and C, presented in the previous section (Section 4)

5.1. Description and Discussion of the Results of Experiment A (Evaluation of the Accuracy of Our Model)

The objective of this experiment was to measure the accuracy of our forecasting model. We considered 15 stocks, covering 3 sectors. Table 11 shows the accuracy of our model for the 15 considered socks. The column, Average Our Model, is the accuracy of forecasting a stock trend by our model. The Average Our Model was calculated by dividing the sum of each five stock accuracies belong to a sector by the total number of stocks in that sector. In each column, we have listed the highest accuracies in bold, and underlined the lowest accuracies. The column Dummy Forecast was the accuracy of conducting a dummy forecast. The accuracy of dummy forecasting reflects the imbalance in the dataset.

The accuracy of our algorithm ranges between 55% (ABC stock) and 80% (EVTC stock). An accuracy of 80%, in the case of EVTC, is the highest accuracy achieved by our model. In contrast, the accuracy of the dummy forecast ranges between 50% (ABBV, FB, and JD stocks) and 65% (EVTC and AVAL stocks). By comparing our model versus the dummy forecast, we can point out that our algorithm provides better accuracies in all cases, except for the case of AVAL. The Wilcoxon test showed that the test statistic equals two, while the critical value for this experiment was 17. When the test statistic is less than the critical value, the null hypothesis is rejected. Thus, the Wilcoxon test rejected the null hypothesis that the median difference between the two models was zero. The figures in the last column 'Average Our Model,' is the statistical mean of the accuracy of our model when applied to each sector. The highest average achieved by our model was by the Technology Sector with an average of 71%, followed by the Finance and the Healthcare sectors with averages equal to 63%, and 62%, respectively.

Table 11. Accuracy of our forecasting model.

Stock Sector	Stock Symbol	Our Model	Dummy Forecast	Average of Our Model
Healthcare	HCA	65	53	62.8
	WAT	71	52	
	ALC	61	54	
	ABBV	62	<u>50</u>	
	ABC	<u>55</u>	54	
Technology	FB	73	51	**70.8**
	AAPL	64	59	
	BABA	74	60	
	JD	63	52	
	EVTC	**80**	**66**	
Finance	ESNT	67	56	64.4
	GNW	68	54	
	GTY	66	62	
	EVR	62	61	
	AVAL	60	65	

We conclude that our model provides higher accuracies than the dummy forecast in 14 out of the 15 considered stocks. Furthermore, higher accuracy in forecasting stocks by our algorithm exists in the Technology sector. Finally, this experiment is statistically significant by rejecting the null hypothesis using the Wilcoxon signed rank test.

5.2. Description and Discussion of the Results of Experiment B (Evaluation of the Usefulness of International Stock Indices)

In this experiment, we wanted to determine which stock had the highest and least impact on each sector.

Table 12 lists the total number of selected lag features in all of the best chromosomes that contributed to predicting the stock trend over the 12 rolling windows. For instance, in the case of the stock symbol ESNT (shown in the first row), we note that the 72 lags of the DAX column have been selected by the Genetic Algorithm-based feature selection algorithm. In contrast, only 36 lags have been selected based on the autocorrelation of the ESNT. In Sum Finance, the first 288 number indicates the number of times that the S&P 500 was selected by our Genetic Algorithm by each stock in the finance sector over the entire 12 rolling windows. Numbers shown in bold denote the most frequently selected indices, while numbers shown in italics indicates the least selected indices.

From Table 12, the autocorrelated total of 312 for Sum Finance suggests that the stocks belonging to the Finance Sector depend on their own past data rather than stock indices. However, the numbers 288 and 252 in the same row suggest that S&P500 and NIKKEI225 contributed to forecasting stock trends in the Finance sector. In contrast, the number *197*, shown in *italics*, suggests that CAC40 was the least selected stock index compared to other stock indices in the Finance Sector. As for Sum Technology, the most frequently selected feature belongs to the stock index S&P 500, with 178 features. On the other hand, stock CAC40 was the least useful stock index for forecasting, with the total number of selected features of *106*. In the case of the Healthcare Sector, the autocorrelated total of 312 indicates that the stocks in the healthcare sector depend on their own historical price movements. In addition, the DAX was useful for forecasting in the Healthcare Sector, with a total number of selected features of 250.

Table 12. The count of the total number of features selected per stock.

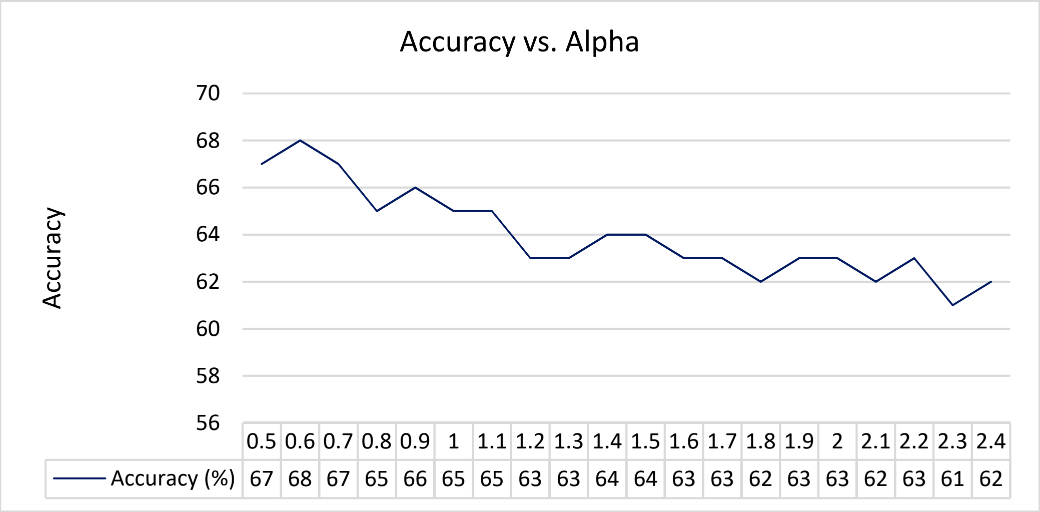
Stock Sector	Stock Symbol	S&P500	NIKKEI225	CAC40	DAX	Auto Correlation
Finance	ESNT	48	48	48	72	36
	GNW	72	12	36	24	108
	GTY	36	72	12	24	12
	EVR	96	108	96	120	120
	AVAL	36	24	5	12	36
Sum Finance		288	264	*197*	252	312
Technology	FB	36	12	4	12	24
	AAPL	60	48	24	59	35
	BABA	3	10	9	11	23
	JD	57	71	59	46	73
	EVTC	22	10	10	9	3
Sum Technology		**178**	151	*106*	137	158
Healthcare	HCA	45	33	6	13	73
	WAT	23	25	24	25	49
	ALC	73	62	83	94	105
	ABBV	45	33	34	81	10
	ABC	35	10	61	37	75
Sum Healthcare		221	*163*	208	250	312
Overall Sum		687	578	*511*	639	782

We conclude from this experiment that

- The historical price movements of a stock can be helpful in predicting stock trends, particularly for the Finance and Healthcare sectors;
- Globally, the results of the last row 'Overall Sum' suggest that the S&P 500 and DAX seem to have a high impact on all the three sectors. The row Sum Finance suggest that the NIKKEI255 had an equal impact on the Finance Sector to the S&P 500. We consider this as an indication that financial markets are closely correlated;
- The last row Overall Sum in Table 12 suggests that overall, the S&P 500 index was the most useful in predicting stock trends. The results in the same row show that CAC40 was the least useful stock index in predicting stock trends.

5.3. Description and Discussion of the Results of Experiment C: Sensitivity Analysis

The objective of this experiment was to determine the accuracy of our model with variations in the threshold value, α, that demarcates Uptrend from Not Uptrend. We re-ran our forecasting model using 20 different values of α. For each value of α, we measured the average accuracy of our model over the 15 selected stocks. The results depicted in Figure 6 provided visual indication that there was a negative correlation between the accuracy of our model and the value of α. The statistical correlation between the two sets of numbers, Accuracy (%) and α, shown at the bottom of Figure 6, was -0.9. This indicates the existence of a strong negative correlation between the accuracy of our model, and the value of α, suggesting that a trader should be cautious while using our model to predict large price changes.



	0.5	0.6	0.7	0.8	0.9	1	1.1	1.2	1.3	1.4	1.5	1.6	1.7	1.8	1.9	2	2.1	2.2	2.3	2.4
Accuracy (%)	67	68	67	65	66	65	65	63	63	64	64	63	63	62	63	63	62	63	61	62

Figure 6. Variation of the accuracy of our mdel as a fucntion of α.

6. Conclusions

6.1. Contribution to Literature

Jiao and Jakubowicz (2017) measured the performance of their forecasting model based on Area Under Curve (AUC) criteria. They reported an average AUC of 0.78. In Experiment A, we computed an average AUC of 0.75. We conclude that their forecasting model outperforms our model. However, there is one major difference between the two studies. Jiao and Jakubowicz (2017) considered more than 200 features, in addition to 8 international stock indices. Therefore, their study cannot provide any insight on the usefulness of just international stock indices to forecast stock trends. In contrast, we considered only four international stock indices. Thus, we consider the accuracy of our model as evidence that international stock indices contribute to forecasting a stock's trend. Another difference is that Jiao and Jakubowicz (2017, p. 20) employed more than 200 technical indicators and concluded that, "Stock movement direction is hardly predictable from its own past data". In this work, based on the results reported in the last row of Table 12, we observed that stock's historical data contributes significantly to forecasting a stock's direction of movement.

6.2. Implications for Stock Trend Forecasting

This paper proposes a forecasting model that predicted if a particular stock exhibited an uptrend with reference to today's closing prices. Our model is a mixture of features selection based on a genetic algorithm and random forest classifier. We have provided evidence that international stock indices can be helpful to forecast stock trends. We considered four international stock indices as the main source for features. We considered the concept of distributed lag analysis and autocorrelation for features engineering. We adopted a genetic algorithm approach to select the most helpful set of features to predict a stock's trend. Finally, we employed the Random Forest algorithm to forecast the next day stock's trend based on the selected set of features.

To examine the performance of our model, we predicted the daily stock trend of 15 stocks from different sectors. The experimental results suggest that the performance of our model significantly outperforms the dummy forecast. In some cases, the accuracy of our model was up to 80%. The results also showed that S&P 500 (the US stock market) is the most useful stock index in the prediction on all sectors. This could be because the 15 considered stocks have been selected from NYSE and NASDAQ. In contrast, CAC40 (French stock index) seems to have the lowest impact on two out of the three sectors. Moreover, we find out that past historical data of the stock itself helps significantly in predicting its trend. However, the results also show that the accuracy of our model decreases considerably while trying to predict large price changes.

6.3. Future Works

In future work, we will test our model with 100 stocks. We accept that a sample of 15 stocks may demonstrate some random effects that overstate or understate stock trends, that would disappear with a larger sample size. Yet, there is a challenge in terms of the lengthy computational time required for a model with 100 stocks, but we will still undertake the test to improve the accuracy of the model.

In addition, our model was tested on monthly basis, but we can do the prediction on a daily basis, which will require high computation. We will work on decreasing the high complexity of the computation, as we extend our work to cover weekly and daily trainings.

Author Contributions: Conceptualization, H.E.-C. and M.E.S.; methodology, A.M.B.; software, A.S.; validation, H.E.-C., R.A. and A.S.; formal analysis, H.E.-C.; investigation, D.J.; resources, S.E.N.; data curation, H.E.-C.; writing—original draft preparation, H.E.-C. and R.A. review and editing, R.A.; visualization, H.E.-C. and A.M.B.; supervision, H.E.-C. and R.A.; project administration, M.E.S. All authors have read and agreed to the published version of the manuscript.

Funding: The research received no external funding.

Data Availability Statement: Data is available from the corresponding author upon request.

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

Note

[1] https://www.investopedia.com/terms/t/transactioncosts.asp (accessed on 10 January 2022).

References

Aloud, Monira E. 2020. Role of attribute selection in a deep ANNs learning framework for high frequency financial trading. *Intelligent Systems in Accounting Finance and Management* 27: 43–54. [CrossRef]

Asadi, Shahrokh, Esmaeil Hadavandi, Farhad Mehmanpazir, and Mohammad Nakhostin. 2012. Hybridization of evolutionary Levenberg-Marquardt neural networks and data pre-processing for stock market prediction. *Knowledge-Based Systems* 35: 245–58. [CrossRef]

Chandan, Kumar, Sumathi Mahadevan, and S. N. Sivanandam. 2016. Prediction of stock market price using hybrid if wavelet transform and artificial neural network. *Indian Journal of Science and Technology* 9: 1–5.

Chen, Sheng. 2018. Stock prediction using convolutional neural network. Paper presented at the IOP conference series on Materials Science and Engineering, Suzhou, China, June 22–24.

El-Chaarani, Hani. 2016. Exploring the Impact of Emotional Intelligence on Portfolio Performance. *Humanomics* 32: 1–28. [CrossRef]

El-Chaarani, Hani. 2019. The Impact of Oil Prices on Stocks Markets: New Evidence During and After the Arab Spring in Gulf Cooperation Council Economies. *International Journal of Energy Economics and Policy* 9: 1–26. [CrossRef]

Fama, Eugene F. 1965. The Behavior of Stock-Market Prices. *Journal of Business* 38: 34–105. [CrossRef]

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

Fama, Eugene F. 1998. Market efficiency, long-term returns, and behavioral finance. *Journal of Financial Economics* 49: 283–306. [CrossRef]

Gidofalvi, Gyozo. 2001. *Using News Articles to Predict Stock*. Working Paper. San Diego: Department of Computer Science and Engineering, University of California.

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]

Hoseinzade, Ehsan. 2019. CNNpred: CNN-based stock market prediction using a diverse set of variables. *Expert Systems With Applications* 129: 273–85. [CrossRef]

Jiao, Yang, and Jeremie Jakubowicz. 2017. Predicting stock movement direction with machine learning: An extensive study on S&&P 500 stocks. Paper presented at the IEEE International Conference on Big Data (BIGDATA), Boston, MA, USA, December 11–14.

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

Karathanasopoulos, Andreas, Mitra Sovan, Chia C. Lo, Adam Zaremba, and Mohammed Osman. 2019. Ensemble models in forecasting financial markets. *Journal of Computational Finance* 23: 101–19. [CrossRef]

Li, Wei, and Jian Liao. 2017. A comparative study on trend forecasting approach for stock price time series. Paper presented at the 11th IEEE International Conference on Anti-Counterfeiting, Security, and Identification (ASID), Xiamen, China, October 27–29.

Mao, Yanan, Zuoquan Zhang, and Dingyuan Fan. 2016. Hybrid feature selection based on improved Genetic Algorithm. Paper presented at the 2016 6th International Conference on Digital Home (ICDH), Guangzhou, China, December 2–4.

Nair, Binoy, S. Ghana Sai, A. N. Naveen, A. Lakshmi, G. S. Venkatesh, and V. P. Mohandas. 2011. A GA artificial neural network hybrid system for financial time series forecasting. *Information Technology and Mobile Communication* 147: 499–506.

Nobel Prize Committee. 2013. Understanding asset prices. In *Nobel Prize in Economics Documents*. Stockholm: Economic Sciences Prize Committee of the Royal Swedish Academy of Sciences.

Prado, Robert. 2011. *The Evaluation and Optimization of Trading Strategy*. Hoboken: John Wiley.

Rahul, Subrat Sarangi, Priyansh Kedia, and Monika. 2020. Analysis of various approaches for stock market prediction. *Journal of Statistics and Management Systems* 23: 285–93. [CrossRef]

Reddy, Vankuru, and Kranthi Sai. 2018. Stock market prediction using machine learning. *International Research Journal of Engineering and Technology (IRJET)* 5: 1033–35.

Rouf, Nusrat, Majid Bashir Malik, Tasleem Arif, Sparsh Sharma, Saurabh Singh, Satyabrata Aich, and Hee-Cheof Kim. 2021. Stock market prediction using Machine Learning techniques: A decade survey on methodologies, Recent developments, and future directions. *Electronics* 10: 2717. [CrossRef]

Sable, Sonal, Ankita Porwal, and Upendra Singh. 2017. Stock price prediction using genetic algorithms and evolution strategies. Paper presented at the International conference of Electronics, Communication and Aerospace Technology (ICECA) Proceedings, Coimbatore, India, April 20–22; pp. 240–53.

Salkind, Neil J. 2010. *Encyclopedia of Research Design*. Thousand Oaks: Sage Publications.

Sang, Chengjie, and Massimo DiPierro. 2019. Improving trading technical analysis with TensorFlow long short-term memory (CSTM) neural network. *The Journal of Finance and Data Science* 5: 1–11. [CrossRef]

Sedighi, Mojtaba, Hossein Jahangirnia, Mohsen Gharakhani, and Saeed F. Fard. 2019. A novel hybrid model for stock price forecasting based on metaheuristics and Support Vendor Machine. *Data, Special Issue on Data Analysis for Financial Markets* 4: 1–20.

Selvin, Sreelekshmy, R. Vinayakumar, E. A. Gopalakrishnan, Vijay Krishna Menon, and K. P. Soman. 2017. Stock price prediction using LSTM, RNN, and CNN-sliding window model. In Paper presented at the 2017 Proceedings of the International Conference on Advances in Computing, Communications, and Informatics (ICACCI), Udupi, India, September 13–16; pp. 1643–47.

Sharma, Ashish, Dinesh Bhuriya, and Upendra Singh. 2017. Survey of stock market prediction using machine learning approach. Paper presented at the International conference of Electronics, Communication and Aerospace Technology (ICECA), Coimbatore, India, April 20–22; Volume 2021, p. 1.

Shen, Jingyi, and M. Omair Shafiq. 2020. Short-term stock market price trend prediction using a comprehensive deep learning system. *Journal of Big Data* 7: 1–33. [CrossRef]

Soni, Payal, Yogya Tewari, and Deepa Krishnan. 2022. Machine Learning approaches in stock price prediction: A systematic review. In *Journal of Physics: Conference Series*. Bristol: IOP Publishing, p. 2161012065.

Sugunnasil, Prompong, and Samerkae Somhom. 2010. Feature selection for neural network based stock prediction. Paper presented at the International Conference on Advances in Information Technology, Bangkok, Thailand, November 4–5.

Tawarish, M., and K. Satyanarayana. 2019. A review on pricing prediction on stock market by different techniques in the field of data mining and genetic algorithm. *International Journal of Advanced Trends in Computer Science and Engineering* 8: 23–26.

Wang, Jie, and Jun Wang. 2015. Forecasting stock market indexes using Principal Component Analysis and stochastic time effective neural networks. *Neurocomputing* 156: 68–78. [CrossRef]

Wang, Wenjian Liu, Linkai Zhu, Rujie Luo, Guang Li, and Shugeng Dai. 2021. Stock price prediction methods based on FCM and DNN Algorithms. *Mobile Information Systems* 2021. [CrossRef]

Wanjawa, Barack Wamkaya, and Lawrence Muchemi. 2014. ANN model to predict stock prices at stock exchange markets. *arXiv* arXiv:1502.06434. Available online: https://arxiv.org/ftp/arxiv/papers/1502/1502.06434.pdf (accessed on 10 January 2022).

Wilcoxon, Frank. 1945. Individual comparisons by ranking methods. *Biometrics Bulletin* 1: 80–83. [CrossRef]

Zhang, Yudong, and Lenon Wu. 2009. Stock market prediction of S P 500 via combination of improved BCO approach and BP neural network. *Expert Systems With Applications* 36: 8849–54. [CrossRef]

Zhanga, Kang, Guoquiang Zhonga, Junyun Donga, Shengke Wanga, and Yong Wanga. 2019. Stock market prediction based on Generative Adversarial Network. *Procedia Computer Science* 147: 400–6. [CrossRef]