Random Forest Regression for Stock Market Prediction

Konstantinos Liagkouras[1] and Konstantinos Metaxiotis[2]
Department of Informatics, University of Piraeus
[1]kliagk@unipi.gr, [2]kmetax@unipi.gr

***Abstract*—This article investigates the challenging task of stock market forecasting. The proposed model is constructed by using Random Forest (RF) regression. The selection of RF regression for addressing this problem is well-justified as improved performance is achieved by merging the predictions from a collection of independent decision trees to generate a more accurate and reliable prediction. Furthermore, RF regression can handle both linear and nonlinear relationships and can capture complex interactions between variables, demonstrating its efficacy in stock market prediction. Empirical research is performed by testing the predictive ability of the proposed RF regression model with real data from Standard and Poor's 500 (S&P500).**

***Index Terms*—Bootstrap sampling, Intelligent Systems, Random Forest, Regression, Stock Market forecasting**

I. Introduction

STOCK market forecasting is a complex task [1] due to the unpredictable nature of financial markets. Stock prices are influenced by a mix of firm specific factors, such as earnings & revenue, dividends [2], debt level and general economic environment factors such as interest rates, inflation, unemployment rate, investments and economic development [3], market sentiment [4], such as news & media coverage, and external events such as political and geopolitical events, which can cause sudden market fluctuations. As a result, financial markets exhibit nonlinear and chaotic behaviour [5], making traditional predictive models less effective.

Despite the fact that recent advances in machine learning [6], [7] and deep learning [8] demonstrate improved stock market forecasting capabilities, these techniques in order to deliver their full potential require a considerable amount of high-quality data. Another challenge is that these techniques are prone to overfitting, reducing the overall performance of the model. As a result, achieving consistently accurate stock market predictions remains an open research question for researchers and practitioners alike.

Random Forest regression is considered appropriate for stock market forecasting. First, random forest has the capability of capturing complex relationships in financial data, such as nonlinear relationships that traditional linear models might fail to identify correctly. Second, random forest can deal well with overfitting due to its ensemble nature. Random Forest by creating many uncorrelated decision trees [9] is able to produce more stable and reliable predictions. Additionally, Random Forest regression [10] can handle effectively missing values and noisy data. Handling well missing values is very crucial as real-world data are often dirty, containing inaccuracies, inconsistencies, or errors [11], [12]. Furthermore, Random Forest can identify the most influential features that affect stock price movement, leading to better investment decisions [13], [14]. Over the past years, the optimal allocation of scarce resources to different investment opportunities [15], [16], [17], [18] has attracted considerable attention from academics and practitioners alike. Below, we review main findings from the available literature in the field.

Du et al. [19] propose a predictive framework for stock markets employing Random Forest techniques. The authors test the proposed model by using two ETF funds. According to the authors the model demonstrates sufficient accuracy aiding decision-making for institutional and individual investors. Tan et al. [20] evaluate the robustness of the random forest (RF) model in the stock selection strategy. The authors train the model by using stocks from the Chinese stock market. The authors examine both fundamental and technical feature space. The authors find evidence of considerable excess returns in the performed out-of-sample testing. Meher et al. [21] experiment with stock forecasting models for top three Fintech Companies of India. They use Random Forest model with high-frequency data in Python. The authors report that the proposed model provides sufficient good predictions as the coefficient of determination of all the selected fintech companies is more than 95%. Lavingia et al. [22] use Random Forest Regression for predicting stock closing prices. The authors use real historical data from the Bombay Stock Exchange (BSE) and National Stock Exchange (NSE) of India to identify optimal entry and exit points for stocks within a specific index. Luong et al. [23] apply the random forest modelling techniques for forecasting the volatility in stock market. The authors

K. Liagkouras is an Assistant Professor at the Department of Informatics, University of Piraeus, (Corresponding author, phone: +30 210 4142276; e-mail: kliagk@unipi.gr).

K. Metaxiotis is a Professor at the Department of Informatics, University of Piraeus, (e-mail: kmetax@unipi.gr).

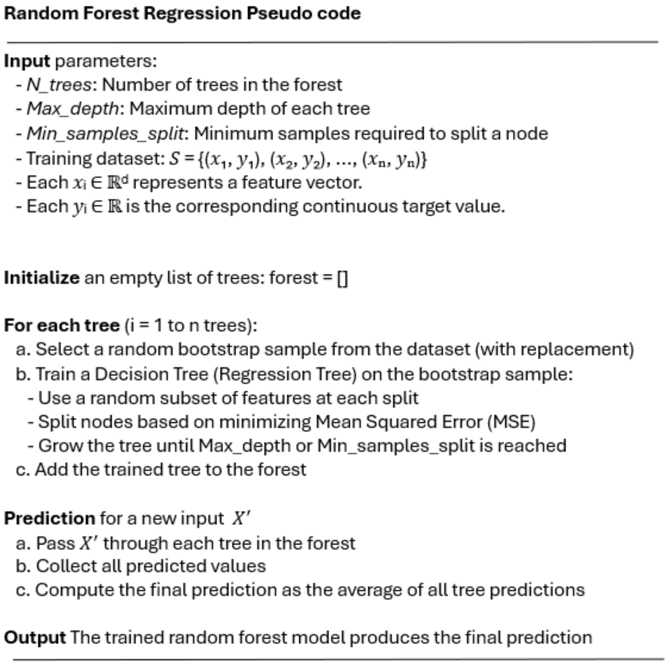
Acknowledgments: This work has been partly supported by the University of Piraeus Research Center

experiment by employing historical intraday data to achieve better forecast accuracy.

The structure of this paper is as follows, section 2 introduces the proposed Random Forest (RF) regression model for stock market forecasting. In section 3, we present the performance metrics and the experimental results of the proposed Random Forest regression model by using real data from S&P500. In Section 4, we provide conclusions and describe possible directions for future work.

II. Random Forest Regression

Random Forest Regression [24] operates by building several decision trees and combining their outputs, yielding more precise and stable forecasting outcomes. Unlike a single decision tree, which can demonstrate instability, as small data changes can cause substantial shifts in the model's predictions, a random forest mitigate variance by combining outputs from many independent trees. The forest comprises multiple decision trees, each trained on a random subset of the dataset by using a bootstrapping technique and considers only a random subset of features at each split, making the model more robust to noise and irrelevant features. Figure 1 illustrates the pseudo code of Random Forest Regression.



```
Random Forest Regression Pseudo code

Input parameters:
  - N_trees: Number of trees in the forest
  - Max_depth: Maximum depth of each tree
  - Min_samples_split: Minimum samples required to split a node
  - Training dataset: $S = \{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}$
  - Each $x_i \in \mathbb{R}^d$ represents a feature vector.
  - Each $y_i \in \mathbb{R}$ is the corresponding continuous target value.

Initialize an empty list of trees: forest = []

For each tree (i = 1 to n trees):
  a. Select a random bootstrap sample from the dataset (with replacement)
  b. Train a Decision Tree (Regression Tree) on the bootstrap sample:
    - Use a random subset of features at each split
    - Split nodes based on minimizing Mean Squared Error (MSE)
    - Grow the tree until Max_depth or Min_samples_split is reached
  c. Add the trained tree to the forest

Prediction for a new input $X'$
  a. Pass $X'$ through each tree in the forest
  b. Collect all predicted values
  c. Compute the final prediction as the average of all tree predictions

Output The trained random forest model produces the final prediction
```

Fig. 1. Random Forest Regression Pseudo code

As shown in Figure 1, the training process for Random Forest Regression [25] follows a few key steps. First, the algorithm selects multiple bootstrap samples from the original dataset. That means that some data points are chosen multiple times, while others may not be included. Then, for each sample with replacement, known as bootstrap sample technique, a decision tree is built by recursively splitting the data based on feature values that minimize the Mean Squared Error (MSE). At each split, only a random subset of features is considered, which helps introduce diversity among trees. Once all trees are trained, the model makes predictions by averaging the outputs of all individual trees, leading to more reliable and unbiased results. Random Forest is one of the more widely used machine learning techniques. Random Forest is an ensemble machine learning method that constructs multiple independent decision trees, each trained on a random subset of the data. Then, each tree generates a prediction outcome in parallel. Finally, we calculate the predicted outcome by calculating the average value of all individual decision trees. During the training phase, multiple samples $s_m = \{(x_1, y_1), \ldots, (x_m, y_m)\}$ are randomly generated by using sample with replacement of the total dataset $s_n = \{(x_1, y_1), \ldots, (x_n, y_n)\}$. This resampling technique is known as bootstrap method. The predicted output of the k-th decision tree is given by the following relationship $f(x,\ s_m^k)$, and the average predicted output ($\hat{y}_{rf}$) of the random forest consisting of k decision trees is given by the following formula:

$$\hat{y}_{rf} = \frac{1}{k}\sum_{i=1}^{k} f(x, s_m^k) \tag{1}$$

Where $\hat{y}_{rf}$ is the Random Forest prediction outcome for the target variable. Figure 2 illustrates the entire process for predicting the outcome by using random forest regression. All data not chosen in the sampling process can be described as out-of-bag data. Finally, the out-of-bag (OB) data can be used for conducting prediction error analysis, as shown by the following relationship:

$$MSE_{OB} = \frac{1}{k}\sum_{i=1}^{k} (\hat{y}_i^{OB} - y_i)^2 \tag{2}$$

Where, MSE_{OB} represents the mean square error of the predicted values on the out-of-bag data, y_i is the actual value, $\hat{y}_i^{OB}$ is the predicted value of the observation y_i in the out-of-bag data, and k is the number of decision trees in the forest.

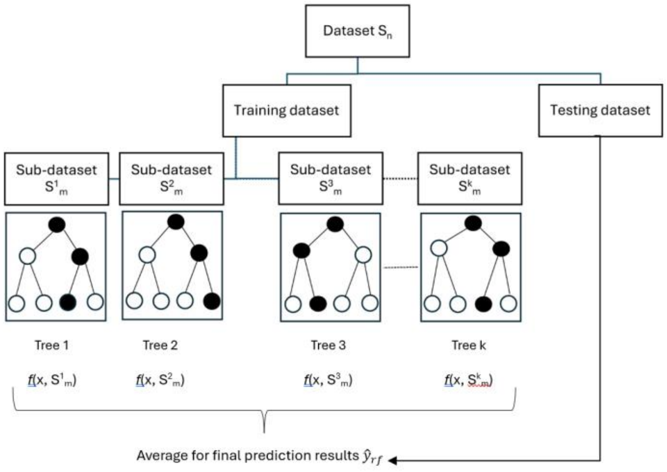


Fig. 2. Random Forest Regression

In ensemble machine learning methods like random forest, out-of-bag (OB) data refers to the subset of training samples not included in the bootstrap sample used to train a particular decision tree. Each tree in a random forest is trained on a random subset of the data with replacement. Therefore, some of the training data is left out of each bootstrap sample. These omitted samples constitute the out-of-bag (OB) data for that tree. The OB data serves as a testing set, for evaluating model's performance.

One of the major advantages of Random Forest (RF) regression is its ability to handle missing data and maintain high accuracy even when dealing with non-linear patterns. However, one drawback is that it requires more computational resources, as multiple trees need to be trained. Figure 3 illustrates the flowchart of the proposed Random Forest (RF) regression model for stock market prediction.

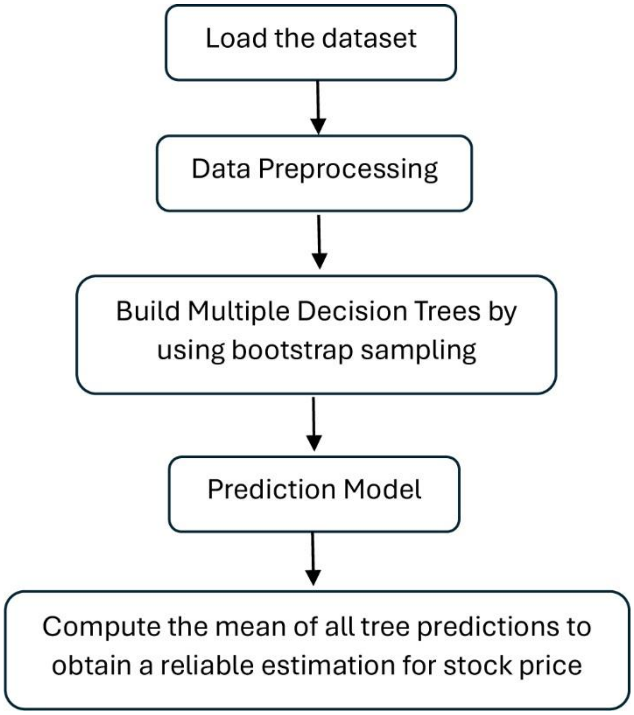

Load the dataset
Data Preprocessing
Build Multiple Decision Trees by using bootstrap sampling
Prediction Model
Compute the mean of all tree predictions to obtain a reliable estimation for stock price

Fig. 3. Flowchart of the proposed Random Forest Regression model for stock market prediction

III. Performance Metrics and Experimental Results

In this section we present the performance metrics for evaluating Random Forest (RF) regression model performance and the corresponding experimental results. The metrics [26] that were utilized by the study are Mean Squared Error (MSE), Root Mean Square Error (RMSE), Mean Absolute Error (MAE) and Mean Absolute Percentage Error (MAPE). In particular, the MSE determines how well the regression line fits a given set of data points. MSE value is directly proportional to the difference between the actual and the predicted output.

$$MSE = \frac{1}{k}\sum_{i=1}^{k}(Actual(y) - Forecast(\hat{y}_{rf}))^2 \qquad (3)$$

Respectively the Root of the Mean Square Error (RMSE) is defined as the square route of the MSE:

$$RMSE(\hat{y}_{rf}) = \sqrt{MSE(\hat{y}_{rf})} \qquad (4)$$

The Mean Absolute Error (MAE) is calculated by averaging the absolute differences between predicted and actual values:

$$MAE = \frac{\sum_{i=1}^{k}\left|Actual(y) - Forecast(\hat{y}_{rf})\right|}{k} \qquad (5)$$

Mean Absolute Percentage Error (MAPE) is defined as the average error produced by the Random Forest regression model and is estimated by the following relationship:

$$MAPE = \frac{1}{k}\sum_{i=1}^{k}\left|\frac{Actual(y) - Forecast(\hat{y}_{rf})}{Actual(y)}\right| \qquad (6)$$

For the experimental evaluation of the Random Forest regression model, we used historical data from Standard and Poor's 500 (S&P500) index. In particular we used the historical data of Amazon (AMZN) from 25th November 2019 to 21st November 2024. The dataset is comprised by 1257 data points. As shown in Figure 2, the 80% of the sample have been used for training, whereas the remaining 20% have been used for evaluating the out-of-sample performance of the Random Forest regression model.

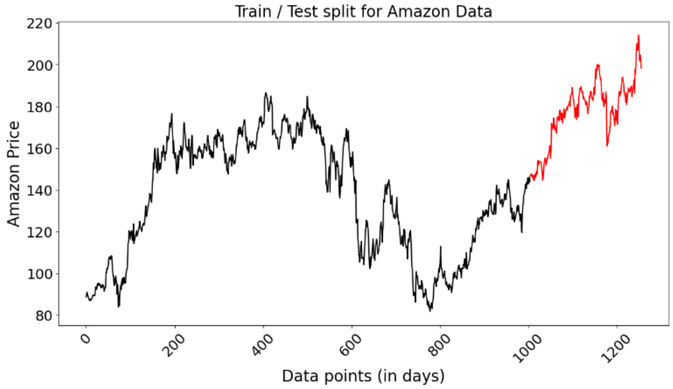


Fig. 4. Training and Testing sample

Table 1 shows the competing configurations of the examined model with 10, 50 and 100 decision trees respectively. As shown in Table 1 the best performance is achieved for 100 decision trees.

TABLE I
RANDOM FOREST REGRESSION: EXPERIMENTAL RESULTS

	Decision Trees	Decision Trees	Decision Trees
	10	50	100
R-squared	78.77%	79.34%	80.39%
MSE	50.86	49.49	46.97
RMSE	7.13	7.03	6.85
MAE	4.60	4.52	4.38
MAPE	2.48	2.43	2.36

Figure 5 shows the actual and predicted stock prices for 100 decision trees. As shown in Figure 5 the Random Forest regression model is not capturing in all cases the underlying relationship between the variables and as a result in certain cases presents poor predictive performance.

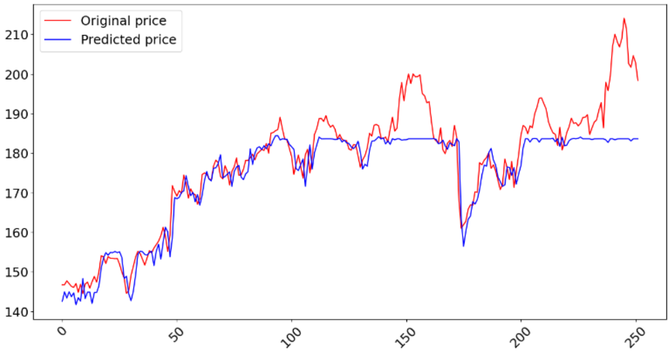


Fig. 5. Real and Predicted values for *100* decision trees

The poor predictive performance of the Random Forest Regression can be attributed to its inability to extrapolate beyond the range of the training data. The source of this problem lies in the fact that Random Forest predicts outcomes by averaging the results of individual decision trees and as a result it cannot generate predictions outside the range of observed target values. This limitation becomes particularly problematic when modeling trends that extend beyond the training data, such as forecasting future values in time series data.

Based on several studies [27], [28] in the field, deep learning techniques such as the Long Short-Term Memory (LSTM) [29] networks outperform Random Forest in stock market forecasting. As these studies suggest, LSTM networks demonstrate superior adaptability to sudden stock price fluctuations, due to their ability to model long-term dependencies and nonlinear relationships in sequential time-series data.

IV. CONCLUSIONS

Random Forest regression is a popular machine learning technique for stock market forecasting. Unlike traditional linear models such as ARIMA [27] or linear regression, which assume a linear relationship between variables, Random Forest can capture complex, nonlinear patterns in stock price movements. Compared to single decision trees [28], Random Forest reduces overfitting by averaging multiple trees, leading to more stable and generalizable predictions. However, on the downside, it lacks the sequential learning ability of models like Long Short-Term Memory (LSTM) [29] networks, which are specifically designed to capture time dependencies in stock price trends. Deep learning models often outperform Random Forest in handling sequential data, however, they necessitate substantial amounts of training data and entail significant computational costs [30]. Support Vector Machines [7] provide strong classification capabilities but may struggle with large, high-dimensional financial datasets. Although Random Forest is a robust and computationally efficient technique, its inability to account for time-series dependencies [31], [32] make it less accurate for long-term stock market forecasting, compared to deep learning techniques [33]. To conclude Random Forest is faster to train and less prone to overfitting than other machine learning techniques, but is often outperformed by more specialized deep learning techniques.

REFERENCES

[1] J. Margaret Sangeetha, K. Joy Alfia, "Financial stock market forecast using evaluated linear regression based machine learning technique", Measurement: Sensors, Volume 31, 2024, 100950.

[2] P. Sinha, A. Kumar, "Do dividends signal earnings quality in the emerging markets? Large sample evidence from India". Int J Syst Assur Eng Manag (2024). https://doi.org/10.1007/s13198-024-02473-x

[3] G. Neidhöfer, M. Ciaschi, L. Gasparini, et al., "Social mobility and economic development", J. Econ. Growth 29, 327–359 (2024). https://doi.org/10.1007/s10887-023-09234-8.

[4] K. Liagkouras, K. Metaxiotis, "Extracting Sentiment from Business News Announcements for More Efficient Decision Making". In: Tsihrintzis, G.A., Virvou, M., Doukas, H., Jain, L.C. (eds) Advances in Artificial Intelligence-Empowered Decision Support Systems. Learning and Analytics in Intelligent Systems, vol 39. Springer, 2024, Cham. https://doi.org/10.1007/978-3-031-62316-5_11

[5] M. Lampart, A. Lampartová and G. Orlando "On risk and market sentiments driving financial share price dynamics". Nonlinear Dyn. 111, 16585–16604 (2023). https://doi.org/10.1007/s11071-023-08702-5.

[6] G.D.C. Cavalcanti, W. Mendes-Da-Silva, I.J. dos Santos Felipe, et al. "Recent advances in applications of machine learning in reward crowdfunding success forecasting". Neural Comput & Applic 36, 16485–16501 (2024). https://doi.org/10.1007/s00521-024-09886-6

[7] K. Liagkouras, K. Metaxiotis, "Stock Market Forecasting by Using Support Vector Machines". In: Tsihrintzis, G., Jain, L. (eds) Machine Learning Paradigms. Learning and Analytics in Intelligent Systems, vol 18. Springer, 2020, Cham. https://doi.org/10.1007/978-3-030-49724-8_11

[8] J. Lederer, "Deep Learning. In: A First Course in Statistical Learning. Statistics and Computing", Springer, 2025 Cham. https://doi.org/10.1007/978-3-031-30276-3_8.

[9] A. Zollanvari, "Decision Trees. In: Machine Learning with Python". Springer, 2023, Cham. https://doi.org/10.1007/978-3-031-33342-2_7

[10] T.T. Tran, N.Q. Phan, H.X. Huynh, "Random Forest Model Parameters Optimization". In: Thai-Nghe, N., Do, TN., Benferhat, S. (eds) Intelligent Systems and Data Science. ISDS 2024. Communications in Computer and Information Science, vol 2191. Springer, Singapore. https://doi.org/10.1007/978-981-97-9616-8_19

[11] K. Liagkouras, K. Metaxiotis, "An Experimental Analysis of a New Interval-Based Mutation Operator", International Journal of Computational Intelligence and Applications 2015 14:03

[12] K. Liagkouras, K. Metaxiotis, "Handling the complexities of the multi-constrained portfolio optimization problem with the support of a novel MOEA". Journal of the Operational Research Society, 69(10), 1609–1627. (2017) https://doi.org/10.1057/s41274-017-0209-4

[13] K. Liagkouras, K. Metaxiotis, “Improving multi-objective algorithms performance by emulating behaviors from the human social analogue in candidate solutions”, European Journal of Operational Research, Volume 292, Issue 3, 2021, pp. 1019-1036.

[14] K. Liagkouras, K. Metaxiotis, “Re-Examining the Optimal Routing Problem from the Perspective of Mobility Impaired Individuals”. In: Tsihrintzis, G.A., Virvou, M., Esposito, A., Jain, L.C. (eds) Advances in Assistive Technologies. Learning and Analytics in Intelligent Systems, vol 28. Springer, 2022, Cham. https://doi.org/10.1007/978-3-030-87132-1_9

[15] K. Liagkouras, K. Metaxiotis, “Multi-period mean–variance fuzzy portfolio optimization model with transaction costs”, Engineering Applications of Artificial Intelligence, Vol. 67, 2018, pp. 260-269.

[16] K. Liagkouras, “A new three-dimensional encoding multiobjective evolutionary algorithm with application to the portfolio optimization problem”, Knowledge-Based Systems, Vol. 163, 2019, pp 186-203.

[17] K. Liagkouras, K. Metaxiotis, “Improving the performance of evolutionary algorithms: a new approach utilizing information from the evolutionary process and its application to the fuzzy portfolio optimization problem”. Ann Oper Res 272, 119–137, 2019.

[18] K. Liagkouras, K. Metaxiotis, and G. Tsihrintzis, “Incorporating environmental and social considerations into the portfolio optimization process”. Ann Oper Res 316, 1493–1518, 2022.

[19] S. Du, D. Hao and X. Li “Research on stock forecasting based on random forest”, 2022 IEEE 2nd International Conference on Data Science and Computer Application (ICDSCA), Dalian, China, 2022, pp. 301-305, doi: 10.1109/ICDSCA56264.2022.9987903.

[20] Z. Tan, Z. Yan and G. Zhu, “Stock selection with random forest: An exploitation of excess return in the Chinese stock market”, Heliyon, Volume 5, Issue 8, 2019, e02310

[21] B.K. Meher, M. Singh, R. Birau and A. Anand, “Forecasting stock prices of fintech companies of India using random forest with high-frequency data”, Journal of Open Innovation: Technology, Market, and Complexity, Volume 10, Issue 1, 2024, 100180.

[22] K. Lavingia, P. Khanpara, R. Mehta, K. Patel and N. “Kothari Predicting Stock Market Trends using Random Forest: A Comparative Analysis”, 2022 7th International Conference on Communication and Electronics Systems (ICCES), Coimbatore, India, 2022, pp. 1544-1550, doi: 10.1109/ICCES54183.2022.9835876.

[23] C. Luong, N. Dokuchaev, “Forecasting of Realised Volatility with the Random Forests Algorithm”, Journal of Risk and Financial Management 11, no. 4: 61, 2018.

[24] H. Li, “Introduction to Machine Learning and Supervised Learning. In: Machine Learning Methods”. Springer, 2024, Singapore. https://doi.org/10.1007/978-981-99-3917-6_1

[25] Y. Manzali, M. Elfar, “Random Forest Pruning Techniques: A Recent Review”. Oper. Res. Forum 4, 43 (2023). https://doi.org/10.1007/s43069-023-00223-6

[26] S. Tarima, N. Flournoy, “The cost of sequential adaptation and the lower bound for mean squared error”. Stat Papers 65, 5529–5553 (2024). https://doi.org/10.1007/s00362-024-01565-x

[27] A.A. Dar, A. Jain, M. Malhotra, et al. “Time Series analysis with ARIMA for historical stock data and future projections”. Soft Comput. 28, 12531–12542 (2024). https://doi.org/10.1007/s00500-024-10309-w

[28] Z. Liu, “Decision Trees. In: Artificial Intelligence for Engineers”. Springer, 2025, Cham. https://doi.org/10.1007/978-3-031-75953-6_4

[29] S.K. Adari, S. Alla, “Long Short-Term Memory Models. In: Beginning Anomaly Detection Using Python-Based Deep Learning”. Apress, Berkeley, CA, 2024, https://doi.org/10.1007/979-8-8688-0008-5_8

[30] K. Metaxiotis, K. Liagkouras, K., “A fitness guided mutation operator for improved performance of MOEAs”, *2013 IEEE 20th International Conference on Electronics, Circuits, and Systems (ICECS)*, Abu Dhabi, United Arab Emirates, 2013, pp. 751-754, doi: 10.1109/ICECS.2013.6815523.

[31] J. Liang, “Comparison of Price Prediction Based on LSTM, GRU, Random Forest, LSSVM and Linear Regression”. BCP Business & Management, (2023) 38, 341-347.

[32] H. Wu, “Comparison of Random Forest and LSTM in Stock Prediction”. Advances in Economics, Management and Political Sciences. 86. 28-34, (2024), 10.54254/2754-1169/86/20240936.

[33] P. Ghosh, A. Neufeld, J.K. Sahoo, “Forecasting directional movements of stock prices for intraday trading using LSTM and random forests”, Finance Research Letters, Volume 46, Part A, 2022, 102280, ISSN 1544-6123, https://doi.org/10.1016/j.frl.2021.102280.