Time Series Cross-Sequence Prediction

KIRIL KOPARANOV, ELENA ANTONOVA, DANIELA MINKOVSKA, KRASIN GEORGIEV
Department of Programming and Computer Technologies,
Department of Aeronautics,
Technical University of Sofia,
8 Kliment Ohridski blvd., 1000 Sofia,
BULGARIA

Abstract: - In the modern transport industry, vast and diverse information arrays, particularly those including time series data, are rapidly expanding. This growth presents an opportunity to improve the quality of forecasting. Researchers and practitioners are continuously developing innovative tools to predict their future values. The goal of the research is to improve the performance of automated forecasting environments in a systematic and structured way. This paper investigates the effect of substituting the initial time series with another of a similar nature, during the training phase of the model's development. A financial data set and the Prophet model are employed for this objective. It is observed that the impact on the accuracy of the predicted future values is promising, albeit not significant. Based on the obtained results, valuable conclusions are drawn, and recommendations for further improvements are provided. By highlighting the importance of diverse data incorporation, this research assists in making informed choices and leveraging the full potential of available information for more precise forecasting outcomes.

Key-Words: - artificial intelligence, automated environments, financial time series, forecasting, machine learning, model.

Received: November 11, 2023. Revised: May 23, 2024. Accepted: June 23, 2024. Published: July 19, 2024.

1 Introduction

Being inherently stochastic due to various factors, financial time series make accurate predictions challenging. With constant advancements in information technology, especially in the field of calculation speed and data storage, the debate regarding the impact of data volume versus data nature on forecasting accuracy has been largely settled, leading researchers to focus on refining the understanding of individual factors' influence. One of these factors that can be quantified and analyzed is financial news, with the use of appropriate tools for the development of an automated forecasting environment, [1], [2], [3]. Other authors are directing their attention towards data preprocessing techniques and the creation of novel formats and models to enhance prediction reliability, [4]. Constituting a particular investment research field, trading signal analysis is the object of interest of scientists who employ hybrid approaches, combining technical analysis with machine learning tools, [5]. Time series forecasting plays a valuable role in numerous domains, such as water purification, where it enables the innovation and expansion of real-time automatic control systems, resulting in significant energy savings, particularly due to their swift operations, [6]. Additionally, it is worth noting that the accumulated expertise from time series forecasting in the realm of securities trading can be effectively utilized in the planning and management of diverse resources, for example, computing and communication. Particularly, it can aid in the migration, timely redirection, and allocation of virtual machines to physical ones, [7]. Within modern society, the demand for stable, continuous, error-free, and resource-efficient systems is principally pronounced in the energy industry. It is logical and expected to exploit the latest technological achievements in these specific economic areas. For instance, in wind farms, where control automation and failure prevention are crucial, secure, user-friendly, and minimally user-dependent time series forecasting plays a vital role, [1], [8]. It also significantly contributes to addressing the pressing issue of container throughput at ports, which has arisen because of the rearrangement of logistics chains brought about by the COVID-19 pandemic, [9].

The accuracy and dependability of time series predictions are pivotal concerns that underpin numerous processes in modern society, [10]. Some authors have started using components from the artificial neural network toolkit in conjunction with other models to make advancements in addressing related issues, [10], [11], [12]. Other researchers are dedicating their efforts to expanding and enhancing already existing environments for automated prediction, [2]. Alternatively, they are conducting experiments to scrutinize outcome precision, [13]. Another research field involves the creation of hybrid models, often incorporating diverse financial data of various origins, [12].

The importance of having efficient and user-friendly time series forecasting methods that do not require extensive training is underscored. In response to this need, automated forecasting environments have emerged in the market, prompting the requisite for a comprehensive evaluation of their performance under different data types. By examining the data from multiple indicators fed into the forecasting system, this study aims to assess the results, draw conclusions, and provide recommendations. Due to the lack of research demonstrating how the nature of data affects prediction accuracy, this paper focuses on revealing the outcomes of replacing the original data sequence with a distinct, yet similarly structured one during the model's training phase. Financial time series were chosen for this purpose, as their intrinsic complexity makes future value predictions challenging. Moreover, they authentically showcase the model's accuracy, thus underscoring the relevance of the approach. Machine learning tools offer numerous beneficial prospects in the fields of investment and statistical analysis, [14]. Prophet was selected as the experimental environment due to the rapid adoption of automated forecasting environments in recent years. Its swift establishment as a trustworthy instrument, [15], [16], [17], along with its ability to predict time series values for relatively long future periods, [18], can aid investors in making well-informed decisions about profitable investments, [19].

It can be argued that automated forecasting environments are not viewed as a research tool, but as one to aid end-user forecasts. The novelty of the applied approach is that new knowledge is acquired about the quality of their activity under different conditions, and this is the way to expand the scope and improve their functionality. All this would lead to an increase in the reliability of forecasting with the use of these environments, as well as in the accessibility of these products to a wider range of users.

2 Methodology

The experiment consists of submitting data into the automatic forecasting system based on Prophet, [1] to determine the impact of each feature (indicator). It was assumed that automatic forecasting systems, such as Prophet, are good enough. In addition, the input data and the model were restricted to a single-time series with no additional features. In this context, the conventional assumption for the univariate analysis that the past values of a variable are the best predictor of its future behavior was challenged.

The used real-world time series data is obtained from Yahoo Finance - daily quotations for The Boeing Company. It contains 14,465 records, from January 2, 1962, to June 19, 2019. Figure 1 illustrates the "Open" price quotations. The input data is divided into sets for training and testing, set in a 4:1 ratio, as visualized and labeled in Figure 1 once again.

Table 1 displays the statistical characteristics of the four distinct features ("Open", "High", "Low", and "Close") within the data set. It is evident that these features exhibit a considerable level of proximity, given the extensive duration of the time series, spanning nearly 60 years.

Table 1. Statistical Characteristics of The Input Data

Characteristic	Open	High	Low	Close
mean	42.735	43.187	42.276	42.748
std	66.868	67.551	66.168	66.897
min	0.38272	0.39095	0.38272	0.38272
max	446.01	446.01	440.19	440.62
25%	2.2798	2.3004	2.2551	2.2798
50%	19.250	19.438	19.063	19.25
75%	54.700	55.180	53.900	54.630

Prophet was preferred as a suitable forecasting platform, as it has had many user installations in recent years. Also, all Python-based software solutions and libraries undergo very fast development, due to the large number of developers and users involved. The Prophet forecasting model, shown in Equation (1) is based on an additive regression model, accounting for trend, seasonality, and holidays, [7]:

$$y(t) = g(t) + s(t) + h(t) + \varepsilon_t, \tag{1}$$

where:

- g(t) - non-periodic trend (piecewise linear or logistic growth curve for modeling non-periodic changes in time series);

- s(t) - periodic changes (e.g., weekly/yearly seasonality), modeled using Fourier series and dummy variables;
- h(t) - holiday effects (user-provided) with irregular schedules;
- ε_t - unusual changes not accommodated by the model (error).

The forecasting model is treated as a curve-fitting problem through probabilistic programming. Therefore, the inherent uncertainty intervals for the trend component are considered. Additionally, the model incorporates "changepoints", allowing the function parameters to change.

Prophet predictions are univariate. This means that the future values of a single variable can be predicted based on the past values of the same variable. Considering the selected data set, the conventional univariate prediction involves only "Close" prices, or only "Open" prices, etc.

In this study, the experiment involved predicting a long target sequence by training the model on historical data from the same sequence (stage one) and from a different sequence (stage two). The former will also be called independent sequence prediction and the latter - cross sequence prediction.

The target/feature sequences considered were the "Open", "High, "Low", and "Close" prices. For example, the model can be trained to predict future "Open" prices using past "Open" prices (denoted as "Open/Open"). Similarly, if it predicts future "High" prices using past "Open" prices, this is denoted as "High/Open".

The performance evaluation of the predictions involves using two metrics - Mean Absolute Error (MAE) and Mean Squared Error (MSE). Their equations (2 and 3) are displayed below:

$$MAE = \frac{1}{n}\sum_{t=1}^{n}|p_t - \hat{p}_t| \quad (2)$$

$$MSE = \frac{1}{n}\sum_{t=1}^{n}(p_t - \hat{p}_t)^2 \quad (3)$$

The parameters in the formulae above are as follows:

- p_t - actual value;
- $\hat{p}_t$ - predicted value;
- n - number of samples

The specific target/feature will be denoted in brackets, e.g. MAE(Open/Open), MAE(Open/High), MAE(High/Open), etc.

The difference in the performances between the different approaches for calculating a target is denoted and calculated as:

MAE_difference(target_i, feature_j) =
MAE(target_i/feature_j) - MAE(target_i/feature_i) (4)

e.g.
MAE_difference(Open, High) =
MAE(Open/High) - MAE(Open/Open)

A closer examination was performed on predictions related to the "Open" prices to confirm the reliability of the outcomes. For this purpose, data from numerous companies was downloaded from Yahoo Finance. Data sets containing more than 5,000 records (similar in size to that of The Boeing Company) were separated for analysis of long sequences, while data sets with fewer than 5,000 entries were used to represent short sequences. Data sets with fewer than 100 records were disregarded.

For each experiment within the set Ξ, a comparison of performance metrics is carried out, and the number of improvements is counted:

$$\text{N\_MAE\_improved}(\text{target}, \text{feature}) = \sum_{\Xi}(\text{MAE\_difference}(\text{target}, \text{feature}) < 0) \quad (5)$$

e.g.

$$\text{N\_MAE\_improved}(\text{Open}, \text{High}) = \sum_{\Xi}(\text{MAE\_difference}(\text{Open}, \text{High}) < 0)$$

All experiments were conducted in Google Colaboratory, a cloud-based integrated development environment (IDE). They were performed using:

2.1 The Python Programming Language (3.10.12)

As a programming language well-known for its readability and simplicity, Python is easy to learn and helpful to developers who want to create a wide range of applications - from web development to data analysis and machine learning. Its adaptability and cross-platform compatibility present it as a great option for creating software that functions smoothly across various operating systems, [20].

2.2 The NumPy library (1.22.4)

This library offers effective data structures and functions for handling large arrays and matrices enabling quick and vectorized operations. It is widely used for numerical computing and scientific computing tasks. Moreover, NumPy is an indispensable part of data analysis, machine learning, and simulation tasks because it easily integrates with other libraries and tools in the scientific Python ecosystem, [21].

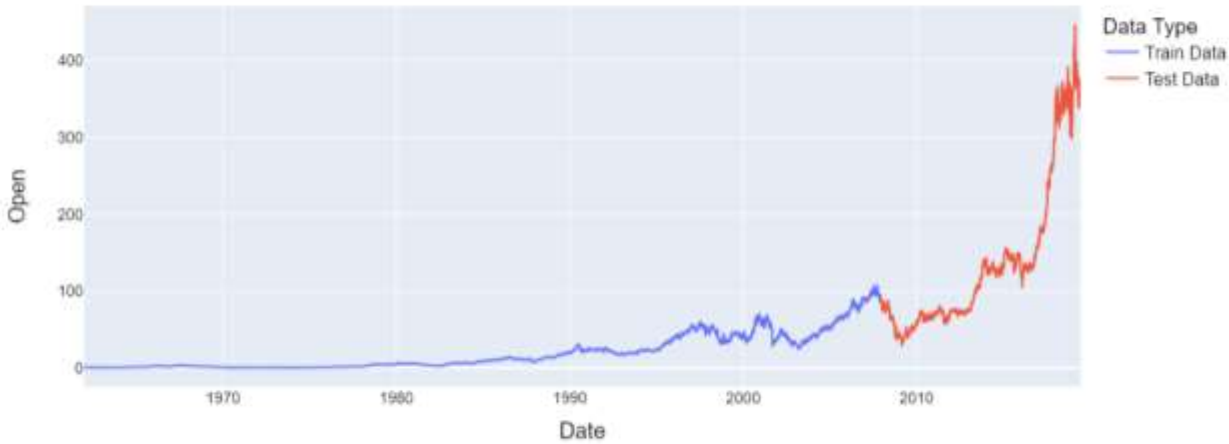

Fig. 1: Input Data Set: “Open” Stock Prices of The Boeing Company (USD)

2.3 The Pandas library (1.5.3)

It can handle and process structured data, as it offers user-friendly data structures and tools for data analysis.

With built-in features such as filtering, sorting, aggregating, and merging, tabular data can be efficiently stored and manipulated using the DataFrame - the fundamental data structure in Pandas. It also provides strong indexing and slicing capabilities for rapid and effective data retrieval, [22].

2.4 The MatPlotLib library (3.7.1)

With this library, plots of all kinds (line, scatter, bar, and histogram) can be quickly generated. Thanks to its wide range of customization choices, users can alter the colors, labels, axes, titles, and other elements of their plots, [23].

2.5 The SkLearn library (1.2.2)

Since it is based on NumPy, SciPy, and MatPlotLib (the libraries mentioned above), it is a strong and extensive library for machine learning tasks - handling missing values, selecting features, and transforming data. While providing a variety of preprocessing methods, SkLearn also offers others for model selection and assessment metrics to evaluate and contrast the performance of various machine-learning models, [24].

2.6 The Prophet procedure (1.1.4)

This procedure meant to work with time series data is based on a decomposable model that includes elements for trend seasonality and holiday effects. It serves as a flexible and user-friendly interface for time series forecasting and is appropriate for both short- and long-term forecasting tasks because it automatically recognizes and models a variety of patterns and outliers in the data provided, [25].

2.7 The Jupyter Notebook

As an open-source web application that facilitates the creation and sharing of documents with live code explanations and visualizations, it aims to make the process of implementing logic and interpreting results faster and more enjoyable. The supported programming languages include Python and R. Code is interactively run in the interface and results are visible right away, [26].

3 Results

The experiment assessment involves utilizing Prophet for automated time series forecasting on a data set comprising daily stock quotes for The Boeing Company. The influence of each feature is evaluated using Equations (2) and (3). The experiment was carried out in two stages. Initially, separate (independent) training and forecasting were performed for each feature (“Open”, “High, “Low”, “Close”) for a period of about 3,000 days into the future. The resulting indicators are presented in Table 2. In the second stage, the error was calculated individually for each feature once more by comparing the predicted prices of the “Open” feature with the actual ones for “Open”, “High”, “Low” and “Close” (cross-training), as shown in Table 3. The performances of both stages are compared in Figure 2. The differences (4) between the indicators of both stages are displayed in Table 4. From there, it is observed that there is no substantial change in prediction accuracy, and the results remain comparable, often slightly better. This suggests that the proposed method leads to some improvement that could substantiate further study. The comparable performances can be attributed to the similarity in nature and values of the input data.

The most favorable outcome is obtained when predicting the “Low” price. This reflection is supported by the results of both performance measures (Table 2 and Table 3), and it is visually represented in Figure 2. Finally, the derived patterns remain consistent across both evaluations.

Table 2. Approximate Mean Absolute Error & Mean Squared Error for the First Stage

	Open/Open	High/High	Low/Low	Close/Close
MAE	54.24	54.85	53.63	54.28
MSE	7707.6	7927.6	7485.4	7718.3

Table 3. Approximate Mean Absolute Error & Mean Squared Error for The Second Stage (Using “Open”)

	Open/Open	High/Open	Low/Open	Close/Open
MAE	54.24	54.88	53.62	54.28
MSE	7707.6	7963.1	7451.9	7719.5

Table 4. Differences between the Results of Both Stages

	Open	High	Low	Close
Mean Absolute Error	0	-0.027	0.009	-0.006
Mean Squared Error	0	-35.50	33.53	-1.227

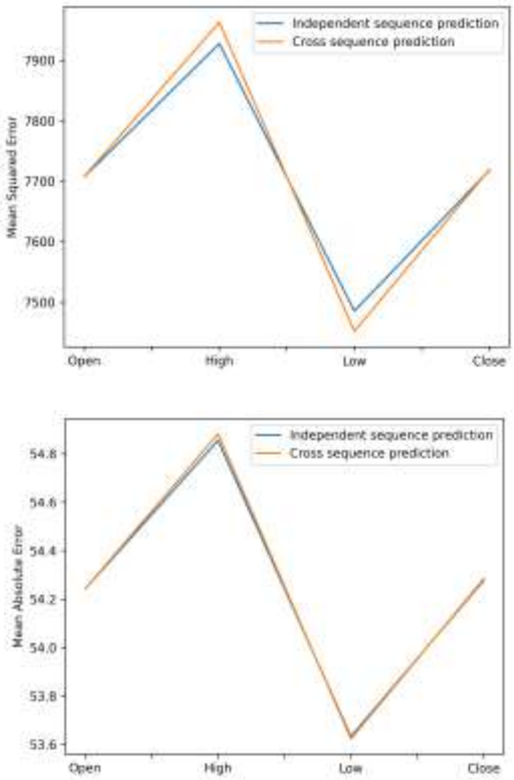

Fig. 2: “Open”, “High”, “Low” & “Close” Price Prediction Errors for The First Stage (Independent Training) & The Second Stage (Cross Training)

Subsequently, the focus shifts to the prediction of solely the “Open” prices. This is achieved by employing models that have been cross-trained on sequences, related to the “High”, “Low”, and “Close” prices. The results are presented in Table 5, which also includes those of the independent prediction (“Open/Open”). The absolute change in the prediction error metric (4), pertaining to the “Open” prices, is visualized in Figure 3.

Table 5. Approximate Mean Absolute Error & Mean Squared Error for the Second Stage (Predicting “Open”)

	Open/Open	Open/High	Open/Low	Open/Close
MAE	54.24	54.22	54.26	54.24
MSE	7707.6	7673.7	7742.8	7706.3

To check the consistency of the findings, the experiment was reiterated using multiple data sets characterized by varying sequence lengths (5010 short and 2165 long sequences). The total experiment count and the instances exhibiting performance improvement (5) are summarized in Table 6. The number of cases with “Open” price prediction error improvement after cross-training with “High”, “Low” and “Close” prices is recorded in the first three rows respectively. We can see that the prediction of “Open” prices based on “Low” prices is better than predictions trained on “Open” prices in 2815 out of 5010 cases with a lower MSE for short sequences and in 1129 out of 2165 cases for long sequences. The number of cases with “Open” price prediction error improvement after cross-training with either “High”, “Low”, “Close” or prices is recorded in the following four rows, named 0, 1, 2, and 3, based on the number of features with improvement. The predictions from either “High”, “Low” or “Close” prices are better in 1797 + 2108 + 550 = 4455 out of 5010 cases for short sequences and in 870 + 985 + 150 = 2005 out of 2165 cases for long sequences.

Figure 4 illustrates a graphical representation of the model’s forecast for the test period (2,893 days into the future), obtained during the initial phase of the experiment. The visualization is as indicated:

- Blue Line: Prediction;
- Black Scatter Plot: Train Data;
- Blue Scatter Plot: Test Data;
- Light Blue Area: Prediction Deviations.

It is noticeable that the trend towards a general, permanent, and continuous increase in price has been captured. In recent decades, however, it has surged significantly. Consequently, fluctuations have also increased, which is not satisfactorily reflected in the forecast. As the forecast duration extends, the expected widening of the confidence interval is observed.

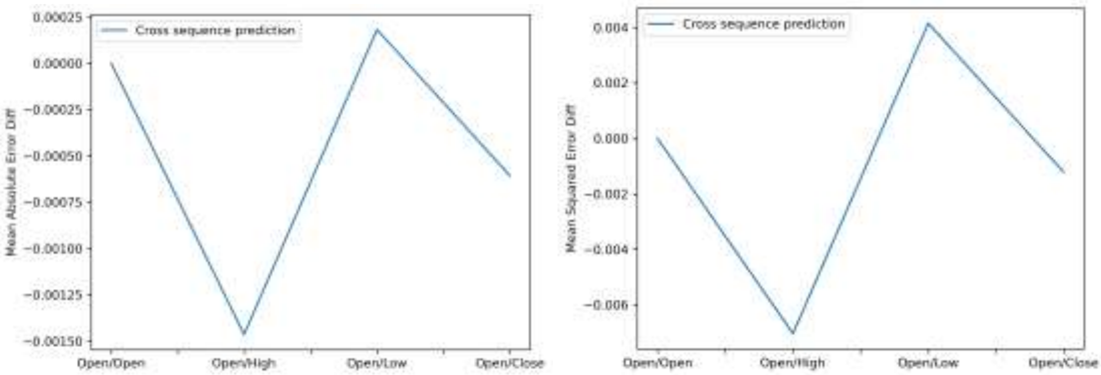

Fig. 3: “Open” Price Prediction Errors for Different Training Sequences (“Open”, “High”, “Low”, “Close”) - Relative to the “Open” Price Prediction

Table 6. Number of Cases with “Open” Price Prediction Error Improvement by Cross-Training & Number of Cases Aggregated by Number of Features with Improvement (“High”, “Low”, and/or “Close”)

	Short Sequences		Long Sequences	
Feature Sequence / Count	**# MAE Reduction**	**# MSE Reduction**	**# MAE Reduction**	**# MSE Reduction**
High	2,226	2,174	1,019	1,023
Low	2,776	2,815	1,131	1,129
Close	2,696	2,674	1,143	1,138
0	528	555	154	160
1	1,817	1,797	876	870
2	2,014	2,108	988	985
3	551	550	147	150
Total	5010		2165	

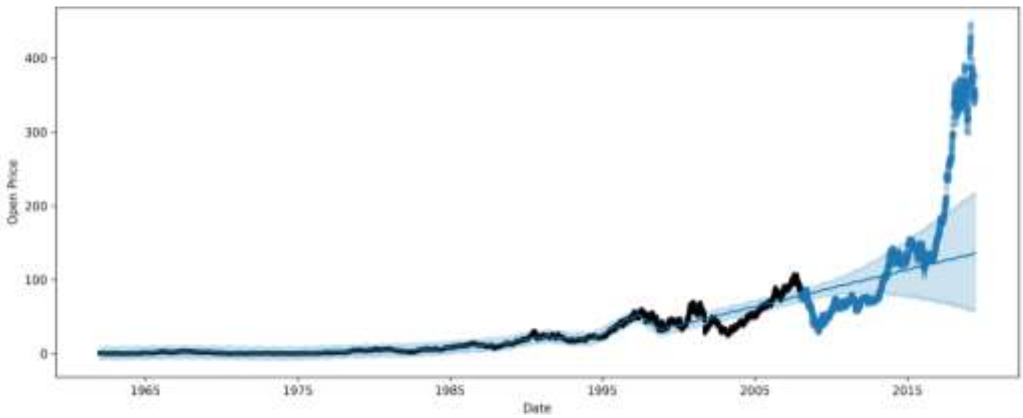

Fig. 4: Example Model Prediction for the Train & Test Periods

4 Conclusion

The conducted experiments and achieved results represent a preliminary stage in knowledge expansion and employment of automated forecasting systems, as well as in considering various factors on their effectiveness. It was demonstrated that by training a model on input sequences that are distinct, yet akin, accurate forecasting is possible. The improvement, although not substantial, remains consistent and warrants additional investigation. In the context of conventional machine learning, a similar concept involves using various attributes of the target. Suggestions can be offered to further diversify the input data by incorporating disparate characteristics.

Acknowledgement:
The authors would like to thank the Research and Development Sector at the Technical University of Sofia for the financial support.

References:

[1] Taylor, S. J., & Letham, B. (2018). Forecasting at Scale. *The American Statistician*, 72(1), 37–45. https://doi.org/10.1080/00031305.2017.1380080.

[2] O. Triebe, H. Hewamalage, P. Pilyugina, N. Laptev, C. Bergmeir, and R. Rajagopal, *"NeuralProphet: Explainable Forecasting at Scale,"* arXiv preprint arXiv:2111.15397, Nov. 2021. https://doi.org/10.48550/arXiv.2111.15397.

[3] S. Mohan, S. Mullapudi, S. Sammeta, P. Vijayvergia and D. C. Anastasiu, "Stock Price Prediction Using News Sentiment Analysis," *2019 IEEE Fifth International Conference on Big Data Computing Service and Applications (BigDataService)*, Newark, CA, USA, 2019, pp. 205-208. https://doi.org/10.1109/BigDataService.2019.00035.

[4] L. Shen, Y. Wei, and Y. Wang, *"Respecting Time Series Properties Makes Deep Time Series Forecasting Perfect,"* arXiv preprint arXiv:2207.10941, Jul. 2022. https://doi.org/10.48550/arXiv.2207.10941.

[5] R. Dash and P. K. Dash, "A hybrid stock trading framework integrating technical analysis with machine learning techniques," *The Journal of Finance and Data Science*, Vol. 2, Issue 1, March 2016, Pages 42-57. https://doi.org/10.1016/j.jfds.2016.03.002.

[6] Kramar V, Alchakov V. Time-Series Forecasting of Seasonal Data Using Machine Learning Methods. *Algorithms*. 2023; 16(5):248. https://doi.org/10.3390/a16050248.

[7] S. Vila, F. Guirado, and J. L. Lérida, "Cloud computing virtual machine consolidation based on stock trading forecast techniques," *Future Generation Computer Systems*, Vol. 145, August 2023, pp.321-336. https://doi.org/10.1016/j.future.2023.03.018.

[8] A. Haghshenas, A. Hasan, O. Osen, and E. T. Mikalsen, "Predictive digital twin for offshore wind farms," *Energy Informatics*, vol. 6, no. 1, pp. 1-26, 2023. https://doi.org/10.1186/s42162-023-00257-4

[9] Z. H. Munim, C. S. Fiskin, B. Nepal, and M. M. H. Chowdhury, "Forecasting container throughput of major Asian ports using the Prophet and hybrid time series models," *The Asian Journal of Shipping and Logistics*, Vol. 39, Issue 2, June 2023, pp.67-77. https://doi.org/10.1016/j.ajsl.2023.02.004.

[10] Y. Huang, Y. Bai, Q. Yu, L. Ding, and Y. Ma, "Application of a hybrid model based on the Prophet model, ICEEMDAN and multi-model optimization error correction in metal price prediction," *Resources Policy*, Vol. 79, Article 102969, Dec. 2022. https://doi.org/10.1016/j.resourpol.2022.102969.

[11] Durairaj, D.M., Mohan, B.H.K. A convolutional neural network-based approach to financial time series prediction. Neural Computing and Applications. S.I.: Deep Learning for Time Series Data, Published: 23 March 2022, Vol. 34, pp.13319–13337. https://doi.org/10.1007/s00521-022-07143-2.

[12] B. H. K. Mohan and M. Durairaj, "Statistical Evaluation and Prediction of Financial Time Series Using Hybrid Regression Prediction Models," *International Journal of Intelligent Systems and Applications in Engineering IJISAE*, 2021, 9(4), 245–255. https://doi.org/10.18201/ijisae.2021473645.

[13] Khekare, G., Turukmane, A.V., Dhule, C., Sharma, P., Kumar Bramhane, L. (2022). Experimental Performance Analysis of Machine Learning Algorithms. In: Qian, Z., Jabbar, M., Li, X. (eds) *Proceeding of 2021 International Conference on Wireless Communications, Networking and Applications. WCNA 2021. Lecture Notes in Electrical Engineering*. Springer, Singapore. https://doi.org/10.1007/978-981-19-2456-9_104.

[14] M. I. A. Aziz, M. H. Barawi, and H. Shahiri, "Is Facebook Prophet Superior than Hybrid Arima Model to Forecast Crude Oil Price?," *Sains Malaysiana* 51(8)(2022): 2633-2643. http://doi.org/10.17576/jsm-2022-5108-22.

[15] I. Ghosh and T. D. Chaudhuri, "Integrating Navier-Stokes equation and neoteric iForest-BorutaShap-Facebook's prophet framework for stock market prediction: An application in Indian context," *Expert Systems with Applications*, Vol. 210, 30 December 2022, 118391. https://doi.org/10.1016/j.eswa.2022.118391.

[16] Aker, Y. (2022). Analysis of Price Volatility in BIST 100 Index with Time Series: Comparison of Fbprophet and LSTM Model. *European Journal of Science and Technology*, Issue 35, 89-93. https://doi.org/10.31590/ejosat.1066722.

[17] Güleryüz, D., & Özden, E. (2020). The Prediction of Brent Crude Oil Trend Using LSTM and Facebook Prophet. *European*

Journal of Science and Technology (20), 1-9. https://doi.org/10.31590/ejosat.759302.

[18] S. Kaninde, M. Mahajan, A. Janghale, and B. Joshi, "Stock Price Prediction using Facebook Prophet," *ITM Web Conf.*, vol. 44 (ICACC-2022, Navi Mumbai, India), p. 03060, May 2022. https://doi.org/10.1051/itmconf/20224403060.

[19] D. Mangesh, V. Sonawane, A. Ehtesham, F. Sharique, and A. Y.S., "Stock Price Prediction using Facebook Prophet," Int. J. Multidiscip. Res., *International Research Journal of Modernization in Engineering Technology and Science*, vol. 5, no. 2, p. 1115-1121, Apr. 2023. https://www.doi.org/10.56726/IRJMETS31379.

[20] Saransh Singh Dangwal. (2023). Python Software Development: Unlocking the Power. DZone, [Online]. https://dzone.com/articles/python-software-development-unlocking-the-power (Accessed Date: July 1, 2024).

[21] Real Python. (n.d.). NumPy Tutorial: Your First Steps into Data Science in Python. Real Python, [Online]. https://realpython.com/numpy-tutorial/ (Accessed Date: July 1, 2024).

[22] Analytics Vidhya. (2023). Navigating Data Formats with Pandas. Analytics Vidhya, [Online]. https://www.analyticsvidhya.com/blog/2023/08/data-formats-with-pandas/ (Accessed Date: July 1, 2024).

[23] Python Star. (2024). How to visualize data with Matplotlib Python Library. Python Star, [Online]. https://www.python-star.com/2024/04/how-to-visualize-data-with-matplotlib.html (Accessed Date: July 1, 2024).

[24] Scikit-learn. (n.d.). Getting Started with Scikit-learn. Scikit-learn, [Online]. https://scikit-learn.org/stable/getting_started.html (Accessed Date: July 1, 2024).

[25] Prophet. (n.d.). Forecasting at scale. Prophet, [Online]. https://facebook.github.io/prophet/ (Accessed Date: July 1, 2024).

[26] Project Jupyter. (n.d.). Jupyter Notebook: The Classic Notebook Interface. Project Jupyter, [Online]. https://jupyter.org (Accessed Date: July 1, 2024).

Contribution of Individual Authors to the Creation of a Scientific Article (Ghostwriting Policy)

The authors equally contributed to the present research, at all stages from the formulation of the problem to the final findings and solution.

Sources of Funding for Research Presented in a Scientific Article or Scientific Article Itself

Funding was provided by the Research and Development Sector at the Technical University of Sofia.

Conflict of Interest

The authors have no conflicts of interest to declare.