International Journal of Artificial Intelligent and Informatics
Vol. 3, No. 2, January 2025, pp. 52~ 59
ISSN: 2622-626X, DOI: http://doi.org/10.33292/ijarlit.v3i1.47

Comparison of LSTM and LSTM with Grid Search Optimization for Stock Price Prediction of Saudi Arabian Oil Company (Aramco)

Eva Yilmaz[1*], Ema Tcrol[2]
[1]École Polytechnique Fédérale de Lausanne, Swiss
[2]University of Rijeka, Croatia
Email: yilmaz.eva@ymail.com

Article Info

Article history:

Received: December 10, 2024
Revised: May 12, 2025
Accepted: July 15, 2025
Available Online: July 30, 2025

Keywords:

LSTM
Grid Search
hyperparameter optimization
stock price prediction
Saudi Aramco

ABSTRACT

This study compares the performance of the Long Short-Term Memory (LSTM) model without optimization and LSTM with Grid Search optimization in predicting Saudi Arabian Oil Company (Aramco) stock prices. Using stock price data from December 2019 to December 2023, this study aims to identify a more accurate prediction model. Results show that the LSTM model with Grid Search optimization provides a significant performance improvement compared to the standard LSTM model, with a decrease in Root Mean Square Error (RMSE) of 11.63% on the test data. This finding indicates the importance of hyperparameter optimization in improving the accuracy of stock price prediction models, especially for the world's largest oil company such as Aramco, whose stock price can be affected by various macroeconomic and geopolitical factors.

1. INTRODUCTION

The stock market plays a crucial role in the global economy and is often an indicator of a country's economic health. As one of the largest oil companies in the world, Saudi Arabian Oil Company (Aramco) has a huge market capitalization and significant influence on global economic stability, particularly in the Middle East region. After conducting an initial public offering (IPO) in December 2019, Aramco's stock became one of the most actively traded in the global market[1] . The volatility of its share price is influenced by various factors, including fluctuations in world oil prices, OPEC+ policies, geopolitical dynamics, and global energy demand, making the prediction of its movements an interesting topic to study.

Stock price prediction is a complex yet very important field in investment management and financial analysis. In recent years, machine learning approaches have shown great potential in predicting stock market movements[2] . One prominent model is the Long Short-Term Memory (LSTM), which is known to capture long-term dependency patterns in time series data, such as stock price fluctuations[3] . However, the effectiveness of LSTM models is highly dependent on the proper selection of hyperparameters, such as the number of units in the hidden layer, dropout rate, learning rate, and batch size. Hyperparameter optimization can significantly improve prediction accuracy, and one of the widely used methods for this purpose is Grid Search, which systematically evaluates various hyperparameter combinations . [4], [5]

Previous research has shown the superiority of LSTM over traditional models such as Random Forest and Logistic Regression in predicting stock movements[6] . Some studies even report prediction accuracy above 95% when using LSTM[7]–[9] . However, hyperparameter optimization remains a challenge, with some studies comparing the effectiveness of Grid Search against other methods such as Random Search[10], [11] . On the other hand, studies specifically on the prediction of stock prices of oil companies, such as those by Sagheer & Kotb (2019)[12] and Saidi et. Al (2020)[13] show that LSTM can identify price movement patterns well, especially when the hyperparameters are optimized. However, no study has specifically tested the performance of LSTM with Grid Search optimization on Aramco stocks, so this study aims to fill that gap.

This research focuses on three main aspects. First, it evaluates the performance of the standard LSTM model (without optimization) in predicting Aramco's stock price. Second, analyzing the impact of Grid Search optimization on improving the prediction accuracy of the model. Third, identifying the best hyperparameter combination for the LSTM model in the context of stock prediction of global-scale energy companies such as Aramco. The findings of this research are expected to provide empirical contributions to the development of stock price prediction models, especially in the application of deep learning for energy commodity stocks. In addition, the results can serve as a reference for investors and financial analysts in making more informed decisions.

2. METHOD

2.1. Data and Data Sources

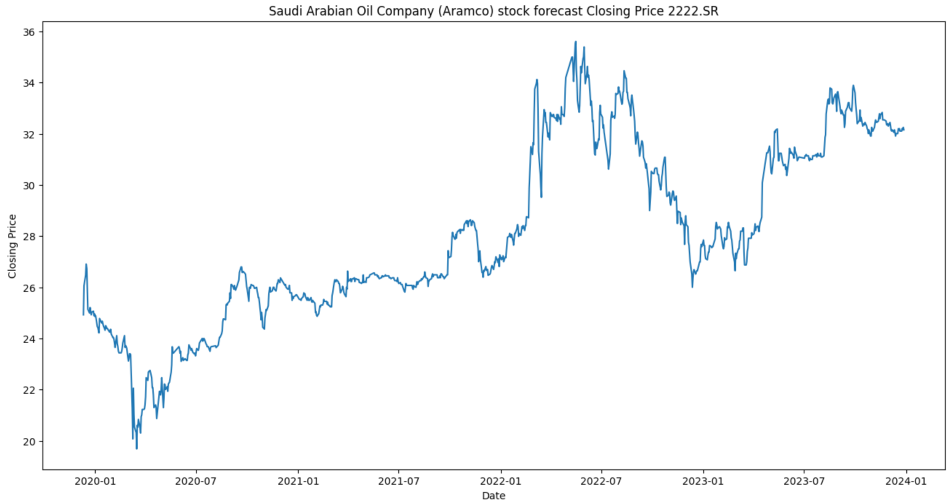

Figure 1. Historical data chart of Aramco stock movements

This study uses Saudi Arabian Oil Company (Aramco) daily stock price data obtained from Yahoo Finance for the period December 11, 2019 (IPO date) to December 31, 2023. The data used includes the opening price (Open), highest price (High), lowest price (Low), closing price (Close), and trading volume. The main focus of the research is on predicting the closing price of Aramco shares.

The dataset is divided into two parts: training data and testing data with a proportion of 80:20. The training data is used to train the LSTM model, while the testing data is used to evaluate the performance of the model in stock price prediction.

2.2. Data Preprocessing

Data preprocessing is an important step in the data analysis process, involving several steps, each of which aims to improve data quality and ease further analysis. The three main stages discussed here are data cleaning, data normalization, and data sequence generation.

2.2.1. Data Cleaning

Data cleaning is a crucial initial stage to remove missing values and outliers. By removing invalid or extreme values, we ensure that subsequent analysis is more accurate and reliable. Research shows that clean data can significantly improve the performance of data analysis models[14] . For example, in the context of meteorology, clean data from SEVIRI meteosats is critical to the nowcasting process, where the quality of the initial data is directly linked to the outcome of weather predictions .[14]

2.2.2. Data Normalization

After data cleaning, normalization is required to ensure that all features are distributed in the same range. One commonly used method is the Min-Max Scaler, which rescales the data so that it falls within a range[14], [15] . This normalization process helps in reducing the scale of differences between features, which can speed up the convergence of machine learning algorithms[16] . In the context of data modeling, especially regarding time series, normalization is very important as it removes biases caused by different scales between data series, improving the overall performance of the model .[17]

2.2.3. Sequence Data Generation

The next stage is the creation of sequence data. In this context, the historical data is converted into a sequence format with a window size of 60 days. This means that to predict the price on day 61, the previous

60 days of data are used as input. This format allows the model to capture temporal patterns in the data, which is very important in time analysis[16] . For example, the use of gated recurrent neural networks (RNNs) in time series analysis has proven to be effective in handling irregular data problems and filling in missing values by utilizing the sequence structure .[16]

2.3. LSTM Model Architecture

Table 1. LSTM Model Architecture

Layer (Sequence)	Layer Type	Main Parameters	Output Dimension	Function
1	LSTM	units=units return_sequences=True input_shape=(60, 1)	(batch_size, 60, units)	Receives a sequence of historical stock price data and learns short-term and long-term temporal patterns. Generates sequential output to be processed by the next LSTM layer.
2	Dropout	rate=dropout_rate	(batch_size, 60, units)	Randomly disabling some neurons during training to prevent overfitting and improve model generalization.
3	LSTM	units=units return_sequences=False	(batch_size, units)	Processes the sequence from the first LSTM layer and generates a single vector representation that summarizes the entire sequence. This allows the model to capture more complex temporal patterns.
4	Dropout	rate=dropout_rate	(batch_size, units)	A second regularization layer to further strengthen the model's resistance to overfitting.
5	Dense	units=25	(batch_size, 25)	The fully connected hidden layer transforms temporal features into a more abstract representation, facilitating the mapping to the final prediction.
6	Dense	units=1	(batch_size, 1)	Output layer that produces a single value prediction for the next period's stock price.

This model structure incorporates several key components that are optimized for financial time series data analysis:

1. First LSTM Layer: This layer accepts inputs with dimensions (60, 1), indicating that the model analyzes 60 historical data points sequentially for each prediction. The parameter return_sequences=True allows this layer to generate the complete sequential output required as input for the next LSTM layer. Each LSTM cell contains a gates mechanism (forget, input, and output) that allows the model to selectively retain or remove information, making it highly effective in capturing temporal patterns in stock price data.
2. Dropout regularization: After the first LSTM layer, a Dropout layer with a specified dropout_rate is applied. This regularization mechanism randomly disables a number of neurons during training to prevent overfitting, improve model generalization, and reduce excessive dependencies on certain features in the training data.
3. Second LSTM Layer: The second LSTM layer with return_sequences=False processes the output of the previous layer and produces a single vector representation that summarizes all sequences. This configuration allows the model to capture complex hierarchical patterns in financial time series data.
4. Second Dropout Regularization: To further improve the model's resistance to overfitting, a second Dropout layer is applied after the second LSTM layer.
5. Hidden Layer Fully Connected: The Dense layer with 25 units serves as a hidden layer that transforms the representation learned by the LSTM layer into more abstract features. This layer allows the model to map the learned temporal representation into a feature space that is more suitable for numerical value prediction.
6. Output Layer: The final Dense layer with one unit serves as the output layer that generates stock price predictions. This architecture is optimized for regression tasks, resulting in the prediction of a continuous value that represents the stock price in the next period.

The overall architecture is designed to capture the complex temporal dynamics in stock price data, while the implemented regularization mechanism helps the model maintain a strong generalization ability over never-before-seen data. Parameter units can be adjusted to control the capacity of the model, with optimal values determined through empirical experiments on the datasets used.

2.3. LSTM Implementation Without Optimization

For the LSTM implementation without optimization, the following hyperparameters are used:

Table 2. Basic hyperparameters of LSTM Model

Parameters	Value
Number of units (units)	50
Dropout rate (dropout_rate)	0.2
Learning rate	0.001
Batch size (batch_size)	32

The model was trained using Adam's optimizer with a Mean Squared Error (MSE) loss function. Model performance was evaluated using Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE), coefficient of determination (R^2), and accuracy metrics.

2.4. LSTM Implementation with Grid Search Optimization

For LSTM with Grid Search optimization, the hyperparameter search space is defined as follows:

Table 3. Hyperparameter Search Space of Grid Search Optimization LSTM Model

Parameters	Search Space
Number of units (units)	[50, 100]
Dropout rate (dropout_rate)	[0.2, 0.3]
Learning rate	[0.001, 0.01]
Batch size (batch_size)	[16, 32]

Grid Search performs a systematic search through all possible hyperparameter combinations and evaluates the performance of each using cross-validation. The hyperparameter combination with the lowest validation loss is selected as the optimal configuration.

2.5. Model Evaluation

The performance of both models was evaluated using several metrics to provide a comprehensive perspective:

1. Root Mean Square Error (RMSE): Measures the square root of the average square of the difference between the predicted value and the actual value .[18]
2. Mean Absolute Error (MAE): Measures the average absolute value of the difference between the predicted value and the true value .[19]
3. Mean Absolute Percentage Error (MAPE): Measures the average percentage absolute error relative to the true value.
4. Coefficient of Determination (R^2): Measures the proportion of variation in the dependent variable that can be explained by the independent variable.
5. Accuracy: Calculated as 100% - MAPE, shows the level of model accuracy in percentage form.

Evaluation is performed on training data to assess the model's ability to learn data patterns, and on test data to assess the model's generalization ability to new data that has not been seen before.

3. RESULTS AND DISCUSSION

3.1. Hyperparameter Search Results using Grid Search

The hyperparameter search with Grid Search resulted in the following optimal parameter combinations:

Table 4. Hyperparameter Search Results with Grid Search

Parameters	Value
Number of units (units)	50
Dropout rate (dropout_rate)	0.2
Learning rate	0.01
Batch size (batch_size)	16
Best validation loss	0.000437

Table 5 shows the evaluation results of various hyperparameter combinations during the Grid Search process:

Table 5. Grid Search Results

Units	Dropout	Learning Rate	Batch Size	Validation Loss
50	0.2	0.001	32	0.000957
50	0.2	0.01	16	**0.000437**
50	0.2	0.01	32	0.000560
50	0.3	0.001	16	0.000809
50	0.3	0.001	32	0.000990
50	0.3	0.01	16	0.000568
50	0.3	0.01	32	0.000580
100	0.2	0.001	16	0.000594
100	0.2	0.001	32	0.000872
100	0.2	0.01	16	0.000471
100	0.2	0.01	32	0.000770
100	0.3	0.001	16	0.000712
100	0.3	0.001	32	0.000944
100	0.3	0.01	16	0.001063
100	0.3	0.01	32	0.000691

From the above results, it can be seen that the hyperparameter combination with units=50, dropout_rate=0.2, learning_rate=0.01, and batch_size=16 provides the lowest validation loss of 0.000437, indicating superior model performance compared to other combinations.

3.2. Model Performance Comparison

Table 6 shows the performance comparison between the LSTM model without optimization and LSTM with Grid Search optimization:

Table 6. Model Performance Comparison

Metrics	LSTM Without Optimization		LSTM with Grid Search	
	Training	Testing	Training	Testing
RMSE	0.54	0.43	0.42	0.38
MAE	0.38	0.33	0.26	0.30
MAPE (%)	1.35	1.02	0.93	0.92
R^2	0.9729	0.7120	0.9833	0.7765
Accuracy (%)	98.65	98.98	99.07	99.08

Results show that the LSTM model with Grid Search optimization provides better performance compared to the LSTM model without optimization. On the test data, the LSTM with Grid Search achieves an RMSE of 0.38, which is lower than the LSTM without optimization (0.43), indicating an error reduction of 11.63%. Similarly, the MAPE for LSTM with Grid Search (0.92%) is lower compared to LSTM without optimization (1.02%), indicating an improvement in prediction accuracy.

The R² value for the LSTM model with Grid Search is also higher (0.7765) compared to the LSTM without optimization (0.7120), indicating that the optimized model can better explain the variation in the data. The accuracy of the LSTM with Grid Search model reached 99.08% on the test data, slightly higher than the LSTM without optimization (98.98%).

3.3. Visualization of Prediction Results

Figures 2 and 3 show a visual comparison between the actual price and the predicted price for the LSTM model without optimization and LSTM with Grid Search optimization on the test data.

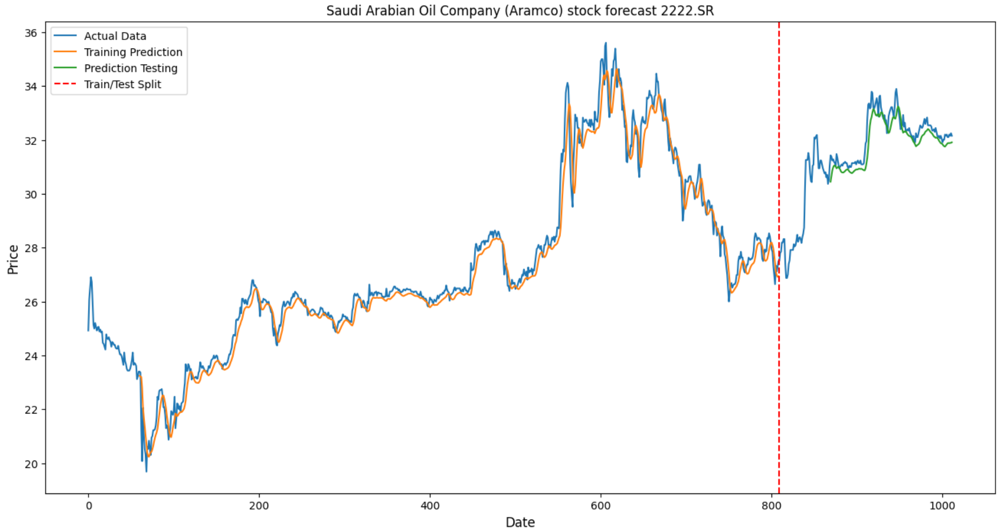

Figure 2. LSTM Prediction Results Without Optimization

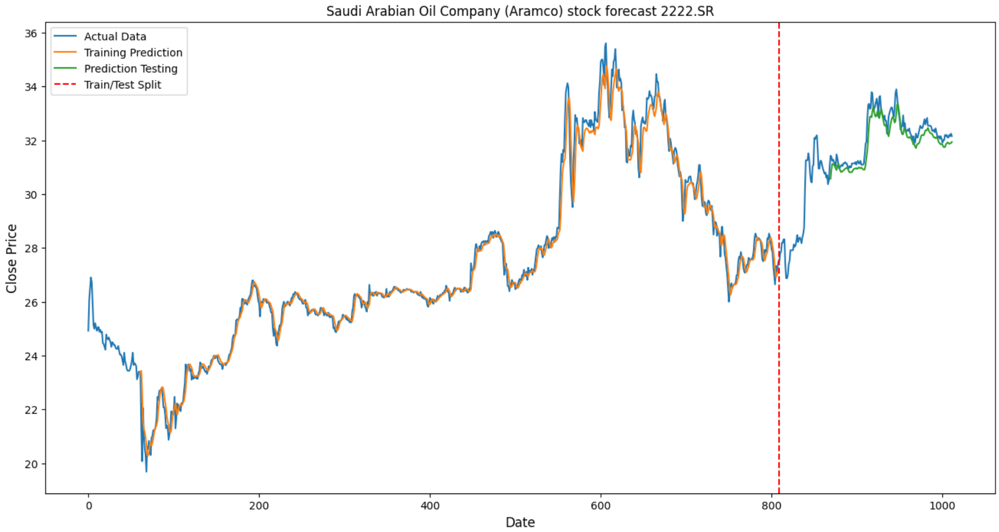

Figure 3. LSTM Prediction Results with Grid Search Optimization

From the visualization, it can be observed that both models perform well in following the trend of Aramco's stock price. However, the LSTM model with Grid Search optimization shows a better match with the actual data, especially during periods of high volatility.

3.4. Analysis and Discussion

The results show that hyperparameter optimization using Grid Search significantly improves the performance of the LSTM model in Aramco stock price prediction. This performance improvement can be explained through several factors:

1. Learning Rate Optimization: Grid Search results show that a higher learning rate (0.01) provides better performance compared to the lower default learning rate (0.001). This indicates that the model requires larger steps during optimization to find a better global minimum.
2. Optimal Batch Size: A smaller batch size (16) proved to be more effective compared to a larger batch size (32). Smaller batch sizes allow for more frequent updates of the model parameters, improving the model's ability to better capture patterns in the training data.

3. Network Architecture: Interestingly, the results show that the model with a smaller number of units (50) is better than the model with a larger number of units (100). This suggests that for the case of Aramco stock price prediction, a simpler model with a smaller number of parameters can reduce the risk of overfitting and provide better generalization.
4. Regularization with Dropout: A lower dropout rate (0.2) results in better performance compared to a higher dropout rate (0.3). This suggests that for the Aramco stock data, a moderate level of regularization is sufficient to prevent overfitting without compromising the model's ability to learn from the data.

This finding is in line with previous research that emphasizes the importance of hyperparameter optimization in deep learning models. Sezer et al. (2020)[2] highlighted that the performance of LSTM models in stock market prediction is highly dependent on the selection of appropriate hyperparameters. Similarly, Poernamawatie et. Al (2024)[20] showed that hyperparameter optimization can significantly improve stock price prediction accuracy.

Interestingly, while both models show very high accuracy (above 98%), the LSTM model with Grid Search optimization still shows consistent improvement across all evaluation metrics. This suggests that even marginal improvements in prediction accuracy can have significant practical implications in the context of stock trading, where small differences in price can translate into substantial gains or losses.

4. CONCLUSION

This study comprehensively compares the performance of the standard LSTM model with the version optimized through Grid Search in predicting the stock price of Saudi Arabian Oil Company (Aramco). The results show that the LSTM model is not only effective (>98% accuracy), but also has significant potential for practical applications in capital market analysis. Hyperparameter optimization was shown to quantitatively improve model performance, with a decrease in RMSE of 11.63% and an increase in R^2 from 0.7120 to 0.7765 on the test data. The identified optimal configuration of 50 LSTM units, dropout 0.2, learning rate 0.01, and batch size 16 indicates that a relatively simple architecture with specific training parameters is suitable for this case. Although the absolute accuracy improvement is marginal, the economic implications are substantial in the context of stock trading, where more precise predictions can have a direct impact on profitability.

REFERENCES

[1] A. Alqahtani, A. Lahiani, and A. Salem, "Crude oil and GCC stock markets: New evidence from GARCH co-integration and Granger causality approaches," *Int. J. Energy Sect. Manag.*, vol. 14, no. 4, pp. 745-756, 2020.

[2] O. B. Sezer, M. U. Gudelek, and A. M. Ozbayoglu, "Financial time series forecasting with deep learning: A systematic literature review: 2005-2019," *Appl. Soft Comput.*, vol. 90, p. 106181, 2020.

[3] S. Siami-Namini, N. Tavakoli, and A. S. Namin, "A comparison of ARIMA and LSTM in forecasting time series," in *2018 17th IEEE international conference on machine learning and applications (ICMLA)*, 2018, pp. 1394-1401.

[4] N. Son, S. Yang, and J. Na, "Deep neural network and long short-term memory for electric power load forecasting," *Appl. Opt. Sci.*, vol. 10, no. 18, p. 6489, 2020.

[5] A. Sinha, T. Khandait, and R. Mohanty, "A Gradient-based Bilevel Optimization Approach for Tuning Hyperparameters in Machine Learning," *arXiv: Learning*. 2020. [Online]. Available: https://arxiv.org/pdf/2007.11022.pdf

[6] T. Fischer and C. Krauss, "Deep learning with long short-term memory networks for financial market predictions," *Eur. J. Oper. Res.*, vol. 270, no. 2, pp. 654-669, 2018.

[7] S. Mehtab and J. Sen, "A Robust Predictive Model for Stock Price Prediction Using Deep Learning and Natural Language Processing," *SSRN Electron. J.*, 2019, doi: 10.2139/ssrn.3502624.

[8] B. Mokona and N. Shipo, "Comparative Analysis of LSTM and Grid Search Optimized LSTM for Stock Prediction: Case Study of Africa Energy Corp. (AFE.V)," *Int. J. Artif. Intell. Informatics*, vol. 2, no. 1, pp. 1-8, 2025, doi: https://doi.org/10.33292/ijarlit.v2i1.30.

[9] J. Muñoz and K. Castaño, "Enhancing Stock Price Predictions: Leveraging LSTM for Accurate Forecasting of Ecopetrol's Stock Performance," *Int. J. Artif. Intell. Informatics*, vol. 2, no. 2, pp. 47-55, 2024, doi: https://doi.org/10.33292/ijarlit.v2i2.36.

[10] T. Hinz, N. Navarro-Guerrero, S. Magg, and S. Wermter, "Speeding up the hyperparameter optimization of deep convolutional neural networks," *Int. J. Comput. Intell. Appl.*, vol. 17, no. 02, p. 1850008, 2018.

[11] H. Naseri, E. O. D. Waygood, B. Wang, and Z. Patterson, "Application of machine learning to child mode choice with a novel technique to optimize hyperparameters," *Int. J. Environ. Res. Public Health*, vol. 19, no. 24, p. 16844, 2022.

[12] A. Sagheer and M. Kotb, "Time series forecasting of petroleum production using deep LSTM recurrent networks," *Neurocomputing*, vol. 323, pp. 203-213, 2019.
[13] L. O. Saidi, H. Aedy, F. Saranani, R. Rosnawintang, P. Adam, and L. O. A. Sani, "Crude Oil Price and Exchange Rate: An Analysis of the Asymmetric Effect and Volatility Using the Non Linear Autoregressive Distributed Lag and General Autoregressive Conditional Heterochedasticity in Mean Models," *Int. J. Energy Econ. Policy*, vol. 10, no. 1, pp. 104-108, 2020, doi: 10.32479/ijeep.8362.
[14] D. Bari, N. Lasri, R. Souri, and R. Lguensat, "Machine Learning for Fog-and-Low-Stratus Nowcasting From Meteosat SEVIRI Satellite Images," *Atmosphere (Basel).*, 2023, doi: 10.3390/atmos14060953.
[15] N. Wulms *et al.*, "The R Package for DICOM to Brain Imaging Data Structure Conversion," *Sci. Data*, 2023, doi: 10.1038/s41597-023-02583-4.
[16] P. B. Weerakody, K. W. Wong, G. Wang, and W. P. Ela, "A Review of Irregular Time Series Data Handling With Gated Recurrent Neural Networks," *Neurocomputing*, 2021, doi: 10.1016/j.neucom.2021.02.046.
[17] M. M. Mokhtar, A. M. Alsamman, H. M. Abd-Elhalim, and A. El Allali, "CicerSpTEdb: A Web-Based Database for High-Resolution Genome-Wide Identification of Transposable Elements in Cicer Species," *PLoS One*, 2021, doi: 10.1371/journal.pone.0259540.
[18] L. Tian, "Forecast and Analysis for Stock Markets of the U.S., Canada, and Mexico Based on Time Series Forecasting Models," *Adv. Econ. Manag. Polit. Sci.*, vol. 13, no. 1, pp. 389-401, 2023, doi: 10.54254/2754-1169/13/20230759.
[19] G. Bekdaş, Y. Aydın, Ü. Işıkdağ, A. N. Sadeghifam, S. Kim, and Z. W. Geem, "Prediction of Cooling Load of Tropical Buildings With Machine Learning," *Sustainability*, vol. 15, no. 11, p. 9061, 2023, doi: 10.3390/su15119061.
[20] F. Poernamawatie, I. N. Susipta, and D. Winarno, "Sharia Bank of Indonesia Stock Price Prediction Using Long Short-Term Memory," *J. Econ. Financ. Manag. Stud.*, vol. 07, no. 07, 2024, doi: 10.47191/jefms/v7-i7-94.