



Forecasting IHSG Stock Prices Using an Attention-Based CNN-BiGRU Hybrid Deep Learning

Ibnu Rasyid Munthe¹, Bhakti Helvi Rambe², Shabrina Rasyid Munthe³

¹Information Systems, Faculty of Science and Technology, Universitas Labuhanbatu, Rantauprapat, Indonesia

²Accounting, Faculty of Economics and Business, Universitas Labuhanbatu, Rantauprapat, Indonesia

³Informatics Engineering, Faculty of Computer Science, Universitas Al Washliyah Labuhanbatu, Rantauprapat, Indonesia

¹ibnurasyidmunthe@gmail.com, ²bhaktihelvirambe@gmail.com, ³shabrinarasyid@gmail.com

Abstract

This study develops an IHSG stock price forecasting model using a hybrid CNN–BiGRU architecture enhanced by an attention mechanism. The key novelty lies in combining CNN-based local pattern extraction with BiGRU-based bidirectional temporal modeling, while attention selectively emphasizes the most informative time steps, improving representation quality for complex and noisy financial series. Historical IHSG data from public sources were preprocessed through feature engineering and normalization, followed by XGBoost-based feature selection to retain the most predictive variables. Model robustness was assessed in two settings: (i) the full dataset and (ii) a “cleaned” dataset excluding the extreme COVID-19 volatility period. The proposed model achieved strong accuracy, with MAE/RMSE of 0.0125/0.02 on the full dataset and 0.0167/0.03 on the cleaned dataset, while Pearson correlation remained close to 1 in both scenarios, indicating high alignment with actual IHSG movements. A 30-day ahead forecast produced a stable and realistic trend. Overall, the CNN–BiGRU with attention provides an effective and robust approach for capturing multi-scale temporal patterns in IHSG forecasting.

Keywords: CNN-BiGRU; financial modeling; IHSG; stock price prediction; time series forecasting

How to Cite: I. R. Munthe, B. H. Rambe, and S. R. Munthe, “Forecasting IHSG Stock Prices Using an Attention-Based CNN-BiGRU Hybrid Deep Learning”, *J. RESTI (Rekayasa Sist. Teknol. Inf.)*, vol. 10, no. 1, pp. 224 - 233, Feb. 2026.
Permalink/DOI: <https://doi.org/10.29207/resti.v10i1.7064>

Received: August 9, 2025

Accepted: February 23, 2026

Available Online: February 28, 2026

*This is an open-access article under the [CC BY 4.0 License](#)
Published by [Ikatan Ahli Informatika Indonesia](#)*

1. Introduction

Stock price prediction remains a classical challenge in the financial domain and continues to be a significant topic among researchers and practitioners [1]. The inherent complexity of the stock market, characterized by its nonlinear, dynamic nature and susceptibility to various economic factors, renders prediction tasks highly challenging [2]. With the increasing availability of multivariate time series data, encompassing historical prices, technical indicators, and trading volumes, there is a growing need for models capable of capturing not only local patterns but also long-term temporal dependencies [3].

Various approaches have been employed to address this issue. Conventional methods such as Support Vector Machines (SVM) [4], Random Forest (RF) [5], and Artificial Neural Networks (ANN) [6] have been widely implemented. However, these models are less effective in capturing the temporal dynamics of financial data. In contrast, deep learning models such as Long Short-

Term Memory (LSTM) and Gated Recurrent Unit (GRU) have demonstrated superior performance in learning sequential patterns from time series data [7], [8]. Additionally, Convolutional Neural Networks (CNN) have proven effective at efficiently extracting local features and have been applied in financial contexts to identify price trends [9]. Recent studies have combined CNN with recurrent architectures, including LSTM and GRU, to create more robust hybrid models [10-13].

The use of hybrid models is considered effective for stock market prediction because they can capture temporal patterns (through LSTM) while simultaneously extracting local patterns and complex features (through CNN). Study [14] using an LSTM–CNN model produced relatively low prediction errors (RMSE 0.1012; MAE 0.0800; MAPE 10.22%) and demonstrated a moderate explanatory capability (R^2 0.4199), indicating that the model was able to capture a portion of data variability while still leaving room for improvement. Meanwhile, another study [15] applied a

Hybrid CNN–LSTM model to the META stock dataset and achieved a MAPE of 0.17995, further confirming that hybrid models can deliver competitive predictive performance on real stock market data, although the accuracy level may vary depending on dataset characteristics, observation periods, and model configurations. Overall, these findings reinforce that hybrid models represent a promising alternative for stock market prediction, particularly when the primary objective is to reduce prediction error and improve model stability in highly fluctuating time-series data.

Nevertheless, several limitations remain in prior stock forecasting studies. Many hybrid models treat all time steps equally and do not include an attention mechanism to highlight the most informative temporal segments, which is suboptimal for long and noisy sequences. In addition, a large portion of existing work relies on univariate inputs or fixed sets of handcrafted indicators, thus underutilizing multivariate information and rarely applying data driven feature selection to reduce redundancy and improve generalization. For the IHSG context, robustness evaluation under extreme volatility regimes such as the COVID 19 shock is also seldom reported, even though such periods can introduce structural breaks and bias model assessment. These gaps motivate an IHSG oriented approach that leverages multivariate modeling with feature importance-based selection such as XGBoost and validates performance under both full data and volatility filtered scenarios.

To address these challenges, this study proposes a hybrid deep learning model that integrates one-dimensional Convolutional Neural Networks (1D-CNN), Bidirectional GRU (BiGRU), and an attention mechanism for multivariate stock price prediction. The CNN component extracts local patterns from input features [12], while the BiGRU captures bidirectional sequential dependencies [16]. The addition of an attention mechanism enables the model to assign varying weights to each time step based on its contribution to the final prediction, thereby enhancing both accuracy and interpretability. Dropout is also employed to prevent overfitting and improve the model's generalization ability [17].

The selection of this architecture is based on the complementary strengths of its components: CNN excels at efficiently extracting features from multivariate inputs, BiGRU comprehensively understands temporal contexts, and attention mechanisms sharpen the model's focus on relevant information. Together, these components form a predictive framework that is not only accurate but also transparent and adaptable to the complexities of financial data. Therefore, the main contribution of this study lies in addressing research gaps in Indonesian stock market forecasting by developing a hybrid model that is more intelligent, interpretable, and relevant to the local market dynamics.

2. Methods

The complete architecture of the proposed hybrid model is illustrated in Figure 1. This model integrates 1D Convolutional Neural Networks (Conv1D), Bidirectional GRU (BiGRU), and an attention mechanism to efficiently capture both spatial and temporal information. Each component is designed to contribute to the representation of multivariate time series data, leading to more accurate and stable stock price predictions. The model structure is visualized hierarchically, from the input layer to the final regression output.

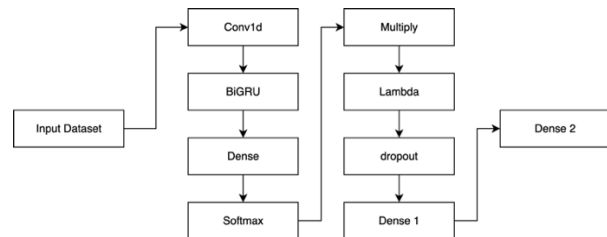


Figure 1. Development Architecture

The following is an explanation of Figure 1.

2.1 Dataset Description

This study utilizes multivariate time series data that reflect stock price dynamics. Each data sample consists of 10 timesteps, with each timestep having five key technical features, namely: open price, high, low, close, and trading volume. The data is processed using a sliding window technique to produce a three-dimensional input format of shape (samples, 10, 5).

All numerical features are normalized using Min-Max Scaling to a range of [0, 1] to speed up the training process and prevent dominance by extreme values. The dataset is split into 80% for training and 20% for testing. Missing values have been removed or filled, ensuring no null values are present during training.

2.2 Architecture Development

The model architecture is built using the Keras Functional API approach and consists of multiple layers designed to capture both spatial (local) and sequential (temporal) patterns, while also assigning attention weights to the most relevant timesteps. The process begins with the InputLayer, which serves as the entry point for the data with an input dimension of (10, 5), meaning the model receives 10 timesteps with 5 features each. This layer has no trainable parameters but defines the input shape for the entire network. The data is then processed by a Conv1D layer with 64 filters, which extracts local patterns from the time series data, such as short-term trends between features or adjacent timesteps [18-20]. The output of Conv1D is a tensor with shape (10, 64), indicating that each timestep is now represented as a 64-dimensional filtered vector. The output of this layer has a shape of (10, 64), indicating that each timestep is represented by a 64-dimensional filtered vector.

Next, the representation is passed to a Bidirectional GRU layer consisting of two directions (forward and backward), each with 64 units, allowing the model to capture long-term dependencies by learning contextual information from both past and future simultaneously [21-24]. The output of this layer is (10, 128), resulting from concatenating both directions (64 + 64). The output of this layer is (10, 128), resulting from the concatenation of both directions (64 + 64). To implement the attention mechanism, the BiGRU output is fed into a Dense layer with a single neuron that computes attention scores for each timestep [25]. Each timestep is assigned an importance value based on the context learned by BiGRU. These scores are then normalized using a Softmax layer to form a probability distribution of shape (10, 1), where the weights sum to one [13]. The result is a vector of shape (10, 1) representing the attention weight for each timestep, summing to 1. The attention weights are applied through a Multiply layer, which performs element-wise multiplication between the BiGRU output (10, 128) and the attention weights (10, 1), producing a weighted tensor (10, 128) that emphasizes more relevant timesteps while reducing the influence of less important ones [26].

The sequence is then summarized using a Lambda layer that applies sum pooling or average pooling across the time dimension [27]. The result is a fixed-dimension vector (128,), referred to as the context vector, which contains global information from the entire attention-weighted sequence, resulting in a fixed-length context vector of shape (128,) that captures global information from the attention-weighted sequence. To reduce overfitting, the context vector passes through a Dropout layer, which randomly deactivates a portion of neurons during training [28], typically with dropout values ranging between 0.2 and 0.5. The representation is then processed by Dense_1, a fully connected layer with 32 units and ReLU activation, which transforms the contextual representation into a denser form and enables the model to learn non-linear relationships before the final prediction. Finally, Dense_2 serves as the output layer with a single unit and linear activation, producing the regression output representing the predicted stock price for the next timestep with an output shape of (None, 1).

Overall, the model contains 54,914 trainable parameters, with the BiGRU layer accounting for the largest portion at 49,920 parameters, indicating that the majority of the model's complexity lies in capturing bidirectional sequential representations.

2.3 Training Configuration

The model is trained using the Adam optimizer with an initial learning rate of 0.001. The Mean Squared Error (MSE) is used as the loss function since the task involves predicting continuous numerical values. Training is conducted for a maximum of 100 epochs with a batch size of 32. An early stopping strategy is applied, where training stops if the validation loss does

not improve for 10 consecutive epochs, preventing overfitting.

2.4 Evaluation Metrics

The model's performance is evaluated using several regression metrics to ensure a comprehensive assessment of prediction accuracy. Mean Absolute Error (MAE) is used to measure the average magnitude of prediction errors by calculating the absolute differences between predicted and actual values, providing an intuitive understanding of overall error levels [29].

Root Mean Squared Error (RMSE) is also employed, which represents the square root of the average squared prediction errors and maintains the original unit of the target variable, making it particularly useful for interpreting the scale of prediction deviations [30]. Additionally, R-squared (R^2) is used to indicate the proportion of variance in the target variable that can be explained by the model, reflecting how well the model captures the underlying data patterns [31].

3. Results and Discussions

This section presents and analyzes the experimental results of the proposed model, namely CNN-BiGRU with attention mechanism. Evaluation was conducted using several standard regression metrics such as Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), Mean Absolute Percentage Error (MAPE), coefficient of determination (R^2) and Pearson Correlation. The model's performance was compared between the training and testing data to observe its generalization capability and to detect potential overfitting. In addition, graphical visualizations and numerical analyses are provided to illustrate the effectiveness of the hybrid architecture in capturing important patterns from multivariate stock price time series data. The discussion is carried out in detail to interpret the model's behavior and its relevance in the context of real-world stock price forecasting.

3.1 Data preprocessing

The data preprocessing step was performed to enhance the quality and information available to the prediction model [32]. This stage includes feature engineering, numerical data transformation, feature selection, and normalization. During feature engineering, several derived features were added to enrich the data representation based on the closing price (Close) and opening price (Open). These derived features include the five-day moving average (MA_5), which reflects short-term trend direction and is commonly used as an early momentum indicator, and the ten-day moving average (MA_10), which provides a more stable trend signal over a longer period. In addition, the percentage daily price change (Price_Change) was calculated by comparing today's closing price with the previous day's closing price to capture short-term movement dynamics. Volatility was computed as the standard deviation of the closing price over the last five days to

represent near-term market risk or instability, while Daily_Return was calculated as the difference between the closing and opening prices on the same day to indicate whether a trading session ended with a gain or loss. These engineered features were introduced to

provide additional information that is not explicitly available in raw data and to help the model better capture short-term patterns, trends, and price fluctuations. The results of preprocessing using feature engineering are shown in Figure 2.

	Date	Close	Open	High	Low	Volume	Price_Change	MA_5	MA_10	Price_Change	Volatility	Daily_Return
0	2019-01-15	6408.78	6347.10	6408.78	6333.64	9.51B	0.115	6341.464	6293.412	0.011468	49.786717	61.68
1	2019-01-16	6413.36	6417.13	6425.94	6385.59	10.43B	0.700	6369.688	6316.630	0.000715	39.714515	-3.77
2	2019-01-17	6423.78	6421.41	6457.73	6410.95	9.92B	0.160	6388.702	6336.907	0.001625	37.907973	2.37
3	2019-01-18	6448.16	6444.21	6452.56	6409.10	8.00B	0.380	6406.040	6354.269	0.003795	41.948476	3.95
4	2019-01-21	6450.83	6448.54	6472.15	6438.90	10.18B	0.400	6428.982	6370.630	0.000414	19.521522	2.29
...	...	...	...	...	...	...	...	...	...	...	...	...
1514	2025-05-22	7166.98	7165.10	7190.67	7136.78	20.12B	0.340	7130.332	7025.848	0.003433	29.364987	1.88
1515	2025-05-23	7214.16	7206.36	7223.25	7177.25	15.22B	0.660	7151.858	7054.641	0.006583	43.568810	7.80
1516	2025-05-26	7188.35	7231.48	7240.08	7129.39	33.10B	-0.360	7161.310	7090.701	-0.003578	45.721930	-43.13
1517	2025-05-27	7198.97	7204.17	7212.57	7162.96	24.54B	0.150	7182.184	7127.318	0.001477	28.068718	-5.20
1518	2025-05-28	7175.82	7223.51	7237.34	7166.68	27.75B	-0.320	7188.856	7146.912	-0.003216	18.650323	-47.69

1519 rows x 12 columns

Figure 2. Preprocessing Results Using Feature Engineering

The preprocessing stage also involved converting the Volume column into numeric values because the original dataset contained string representations with units such as “M” (million) and “B” (billion), which cannot be directly processed by machine learning

models. Therefore, each value was transformed into its full numeric form so that the model could correctly interpret trading activity data; for example, a value such as “6.09M” was converted into 6,090,000. The results of this conversion process are presented in Figure 3.

	Tanggal	Close	Open	High	Low	Volume	Perubahan (%)	MA_5	MA_10	Price_Change	Volatility	Daily_Return
0	2019-01-15	6408.78	6347.10	6408.78	6333.64	9.510000e+09	0.115	6341.464	6293.412	0.011468	49.786717	61.68
1	2019-01-16	6413.36	6417.13	6425.94	6385.59	1.043000e+10	0.700	6369.688	6316.630	0.000715	39.714515	-3.77
2	2019-01-17	6423.78	6421.41	6457.73	6410.95	9.920000e+09	0.160	6388.702	6336.907	0.001625	37.907973	2.37
3	2019-01-18	6448.16	6444.21	6452.56	6409.10	8.000000e+09	0.380	6406.040	6354.269	0.003795	41.948476	3.95
4	2019-01-21	6450.83	6448.54	6472.15	6438.90	1.018000e+10	0.400	6428.982	6370.630	0.000414	19.521522	2.29
...	...	...	...	...	...	...	...	...	...	...	...	...
1514	2025-05-22	7166.98	7165.10	7190.67	7136.78	2.012000e+10	0.340	7130.332	7025.848	0.003433	29.364987	1.88
1515	2025-05-23	7214.16	7206.36	7223.25	7177.25	1.522000e+10	0.660	7151.858	7054.641	0.006583	43.568810	7.80
1516	2025-05-26	7188.35	7231.48	7240.08	7129.39	3.310000e+10	-0.360	7161.310	7090.701	-0.003578	45.721930	-43.13
1517	2025-05-27	7198.97	7204.17	7212.57	7162.96	2.454000e+10	0.150	7182.184	7127.318	0.001477	28.068718	-5.20
1518	2025-05-28	7175.82	7223.51	7237.34	7166.68	2.775000e+10	-0.320	7188.856	7146.912	-0.003216	18.650323	-47.69

1519 rows x 12 columns

Figure 3. Preprocessing results using Volume to Numeric Value Conversion

After all features were prepared, feature selection was carried out using the XGBoost Feature Importance algorithm to measure the relative contribution of each feature to the performance of the initial prediction model [33]. Only features with high importance scores were retained to reduce the risk of overfitting, speed up the training process, and improve the generalization capability of the model [34]. The results of feature selection are illustrated in Figure 4.

```

Close: 0.8328
MA_5: 0.0810
Low: 0.0644
Open: 0.0161
High: 0.0050
Volume: 0.0001
Price_Change: 0.0001
MA_10: 0.0001
Perubahan (%): 0.0001
Volatility: 0.0001
Daily_Return: 0.0000

```

Figure 4. Preprocessing Result using Feature Selection with XGBoost

The final step in preprocessing involved feature normalization using the MinMaxScaler technique, which transforms all feature values into the range [0, 1],

preventing the model from being biased toward features with large numeric values and accelerating convergence during training [35]. This normalization process also ensures that the model can handle multivariate data with different distributions in a balanced manner. The results of the normalization stage are shown in Figure 5.

	Close	Open	High	Low	MA_5	Date
0	0.622807	0.607414	0.603438	0.614446	0.598824	2019-01-15
1	0.623962	0.625068	0.607969	0.627626	0.606403	2019-01-16
2	0.626588	0.626147	0.616364	0.634060	0.611510	2019-01-17
3	0.632732	0.631894	0.614999	0.633591	0.616166	2019-01-18
4	0.633405	0.632986	0.620172	0.641151	0.622327	2019-01-21

Figure 5. Preprocessing Result using Normalization with MinMaxScaler

3.2 Evaluation Result with COVID-19 Spike (Full Data)

The training process of the CNN-BiGRU model with an attention mechanism produced strong results for stock price prediction using multivariate time series data. The evaluation was conducted in several stages, beginning with training visualizations, followed by quantitative

performance measurement on the test set, and concluding with a short-term forecast for the next 30 days. As shown in Figure 6, the loss and Mean Absolute Error (MAE) curves for both training and validation decrease sharply within the early epochs and then gradually stabilize, indicating that the model reaches convergence in a controlled manner. The gap between the training and validation curves remains small, which

suggests that the learning process is stable and that the model maintains consistent performance when exposed to unseen data. To further reinforce this observation, Figure 7 presents a snapshot of the training log near the final epochs (epochs 224 to 229), where the validation loss and MAE remain consistently low, supporting the conclusion that the model training process is steady and optimized.

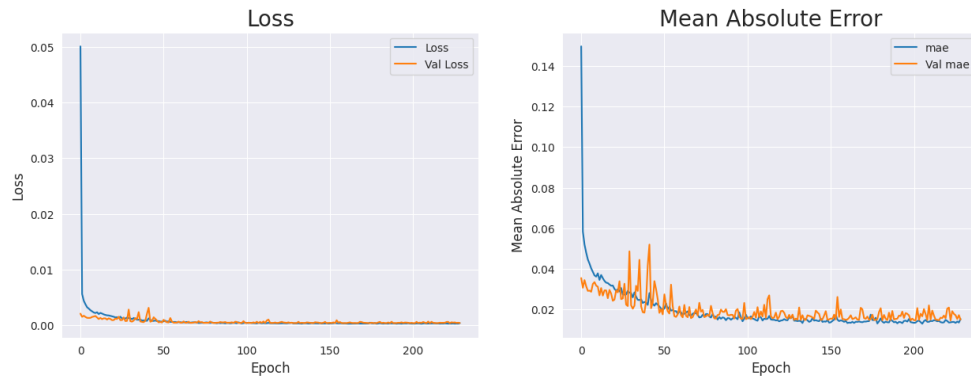


Figure 6. Loss and MAE graphs on training and validation data during the training process

```
Epoch 224/500
38/38 ----- 1s 31ms/step - loss: 2.9885e-04 - mae: 0.0134 - val_loss: 5.5569e-04 - val_mae: 0.0191
Epoch 225/500
38/38 ----- 1s 32ms/step - loss: 3.3244e-04 - mae: 0.0139 - val_loss: 4.9816e-04 - val_mae: 0.0178
Epoch 226/500
38/38 ----- 1s 32ms/step - loss: 3.3313e-04 - mae: 0.0136 - val_loss: 5.3338e-04 - val_mae: 0.0171
Epoch 227/500
38/38 ----- 2s 49ms/step - loss: 3.6613e-04 - mae: 0.0146 - val_loss: 4.4767e-04 - val_mae: 0.0153
Epoch 228/500
38/38 ----- 3s 48ms/step - loss: 3.2508e-04 - mae: 0.0135 - val_loss: 4.7623e-04 - val_mae: 0.0172
Epoch 229/500
38/38 ----- 2s 30ms/step - loss: 4.2326e-04 - mae: 0.0158 - val_loss: 4.3187e-04 - val_mae: 0.0149
```

Figure 7. A snapshot of the model training log on the last epoch showing the stability of the loss and MAE values.

After training, the model was evaluated using 20% of the testing data. The evaluation results in Figure 8 show that the model achieved a MAE of 0.0125, indicating that the predicted values deviate only slightly from the actual values on average. The RMSE value of 0.02 further confirms that the magnitude of prediction errors remains low, while an R^2 value of 0.95 indicates that the model explains 95% of the variance in stock price movements. In addition, the MAPE of 1.31% shows that the relative prediction error is small compared to the true values. Collectively, these metrics suggest that the model performs with high accuracy and stable predictive behavior. The Pearson correlation coefficient between predicted and actual values is 0.9770, which further indicates a very strong linear alignment between model outputs and observed data.

```
10/10 ----- 0s 9ms/step - loss: 2.8827e-04 - mae: 0.0127
10/10 ----- 2s 99ms/step
MAE : 0.02
RMSE : 0.02
R2 : 0.95
MAPE : 1.99%

Pearson Correlation: 0.9754
```

Figure 8. Model evaluation results on test data. MAE, RMSE, R^2 , MAPE and Pearson correlation metrics show high and stable predictive performance.

To visually assess whether the model can follow market movement patterns, a comparison between actual closing prices and predicted prices on the test data is

provided in Figure 9. The figure shows that the prediction curve closely follows the actual price curve over time, capturing upward, downward, and sideways trends with minimal divergence. Both smaller variations and larger fluctuations are reproduced in a manner consistent with the real series, suggesting that the CNN-BiGRU with attention effectively captures market dynamics. The close proximity between both curves is consistent with the earlier quantitative results, particularly the MAE of 0.0125 and Pearson correlation of 0.9770, indicating that prediction errors remain limited across the test period. This visualization strengthens confidence that the model is not only numerically accurate but also capable of visually replicating observed market behavior, which is important for short-term forecasting.

As an additional assessment of short-term forecasting usefulness, the model was used to predict stock prices for the next 30 days using the most recent historical data as input. Figure 10 presents the forecasted values in red as a continuation of the historical closing prices shown in blue. The predicted series exhibits a relatively stable trajectory that remains consistent with the latest direction of the historical trend, without producing unrealistic sharp oscillations. This suggests that the model has learned meaningful temporal patterns from historical movements rather than amplifying noise. Such short-term forecasts can support daily-to-weekly

decision making in trading strategies or mid-term investment planning, although the predictions should still be interpreted cautiously because external factors

and sudden market events may influence real prices beyond what historical patterns can capture.

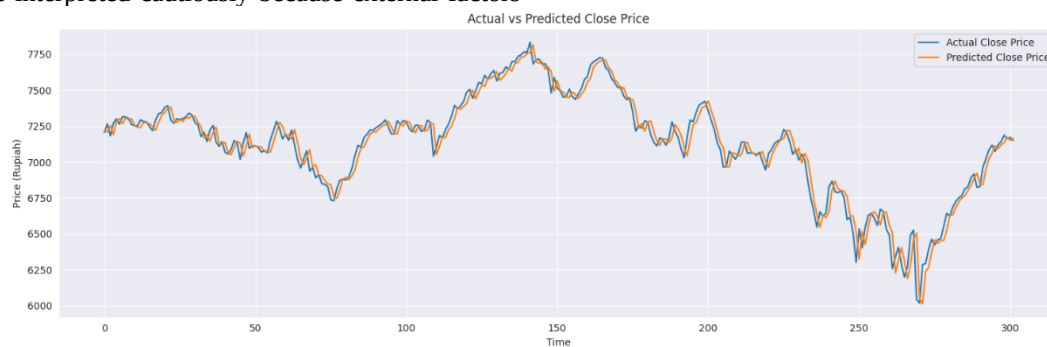


Figure 9. Comparison graph of actual closing prices and model predictions on the test data

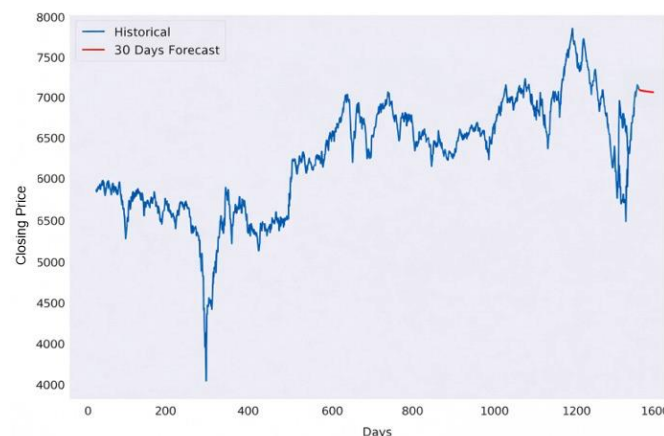


Figure 10. IHSG price prediction for the next 30 days based on historical data

3.3 Evaluation Result without COVID-19 Spike

A follow-up experiment was conducted to evaluate the CNN-BiGRU model with an attention mechanism using a dataset that was cleaned by removing the extreme period associated with the COVID-19 market spike. This additional evaluation aimed to examine whether the model behaves more stably and produces more reliable predictions when the input series is not heavily affected by large-scale global anomalies. The training process on the filtered dataset demonstrates stable learning behavior, as evidenced by the training log in Figure 11, which captures the final training phase (epochs 59 to 64) and shows a consistent decrease in

both training and validation loss as well as MAE. Importantly, there are no clear indications of overfitting, since the validation curves remain low and within a stable range throughout the final epochs. This observation is further supported by the full training visualization in Figure 12, where loss and MAE drop rapidly in the initial epochs and then stabilize, suggesting that the model converges efficiently. Although minor fluctuations appear in the validation MAE, the overall trend remains stable and does not show divergence between training and validation performance, indicating effective learning under the cleaned, more stable market condition.

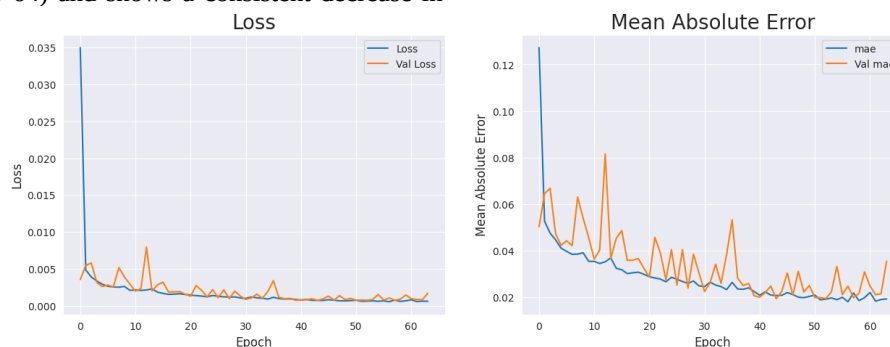


Figure 11. Model training log snippet from the final epoch

After training, the model was evaluated on the test set and achieved solid predictive performance under the filtered data condition. The evaluation metrics in Figure 13 report a MAE of 0.0167, which indicates that the average absolute deviation between predictions and actual values is relatively small. The RMSE of 0.03 likewise confirms that larger errors are limited in magnitude, while the R^2 value of 0.93 shows that the model still explains most of the variance in IHSG price

movements even after removing the extreme COVID-19 period. The MAPE result of 1.58% further supports that prediction errors remain low in relative terms, and the Pearson correlation coefficient of 0.9755 indicates that the predicted series maintains a very strong linear alignment with the actual data. Together, these results suggest that the model continues to perform reliably and remains robust even when trained and tested under calmer market dynamics.

```
Epoch 59/500
37/37 ----- 1s 33ms/step - loss: 5.5195e-04 - mae: 0.0182 - val_loss: 8.7950e-04 - val_mae: 0.0217
Epoch 60/500
37/37 ----- 2s 50ms/step - loss: 7.2775e-04 - mae: 0.0207 - val_loss: 0.0015 - val_mae: 0.0310
Epoch 61/500
37/37 ----- 3s 68ms/step - loss: 0.0010 - mae: 0.0257 - val_loss: 9.5957e-04 - val_mae: 0.0249
Epoch 62/500
37/37 ----- 4s 28ms/step - loss: 5.5655e-04 - mae: 0.0181 - val_loss: 8.5209e-04 - val_mae: 0.0211
Epoch 63/500
37/37 ----- 1s 28ms/step - loss: 5.9256e-04 - mae: 0.0188 - val_loss: 8.1111e-04 - val_mae: 0.0216
Epoch 64/500
37/37 ----- 1s 28ms/step - loss: 5.8448e-04 - mae: 0.0190 - val_loss: 0.0017 - val_mae: 0.0356
```

Figure 12. Loss and MAE graphs during model training without COVID-19 spike data.

```
10/10 ----- 0s 8ms/step - loss: 7.9269e-04 - mae: 0.0222
10/10 ----- 1s 83ms/step
MAE : 0.03
RMSE : 0.03
R2 : 0.93
MAPE : 3.78%

Pearson Correlation: 0.9639
```

Figure 13. Model evaluation results: MAE, RMSE, R^2 , MAPE, and Pearson correlation on data without COVID-19 spikes

To provide a more intuitive assessment of predictive behavior, a visualization comparing actual closing prices and predicted values is shown in Figure 14. The figure indicates that the predicted curve closely tracks the actual curve across nearly the full evaluation horizon, following both upward and downward movements without obvious lag or sharp deviations. This close alignment supports the quantitative results, particularly the low MAE and high Pearson correlation, and suggests that the model can reconstruct historical patterns effectively even when market volatility is lower. In addition to predictive accuracy, the distribution of attention weights was examined to improve interpretability. The attention analysis indicates that higher weights are frequently assigned to recent timesteps, especially those involving the closing price (Close) and short-term moving average features such as MA5, suggesting that the model emphasizes near-term momentum and daily fluctuations as primary

drivers of prediction. In contrast, trading volume and longer time-lagged information tend to receive lower attention scores, indicating a smaller contribution to the final forecast. This behavior is consistent with common financial intuition, where short-term dynamics often dominate near-term market direction, and the attention visualization helps clarify which inputs contribute most strongly to the model's forecasting decisions.

Following the satisfactory performance on the cleaned dataset, the model was further applied to generate an IHSG price forecast for the next 30 days using the latest historical inputs in a rolling forecast manner. The results are presented in Figure 15, where the forecasted values are shown as a red continuation line following the blue historical closing-price series. The model suggests a moderately increasing trend over the 30-day horizon, and the projected trajectory appears consistent with the preceding historical trend without producing unrealistic sharp oscillations. This outcome indicates that the model maintains its generalization capability under more stable market conditions and can produce short-term forecasts that are potentially useful for practical applications such as weekly investment strategy planning, technical analysis support, and simulation of trend-based price movements grounded in historical patterns.

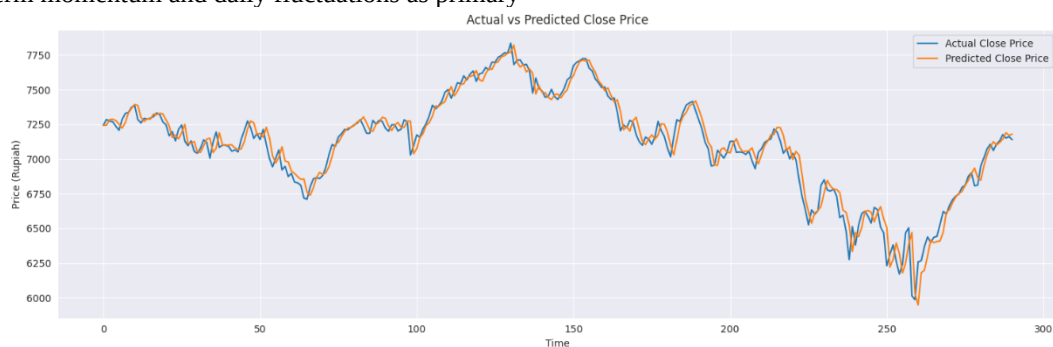


Figure 14. Comparison chart of actual closing prices and model predicted results on test data (without COVID-19 spike)

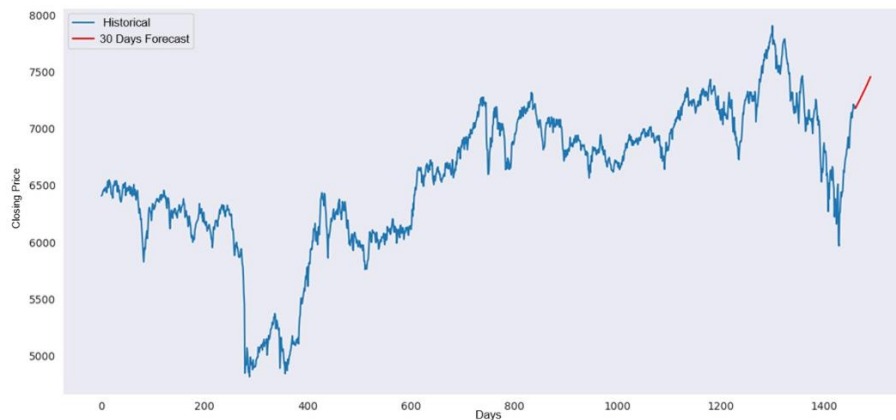


Figure 15. IHS price prediction for the next 30 days based on historical data without a COVID-19 spike

3.2 Result Comparison

Result Comparison explains the impact of extreme market conditions on model performance by comparing two experimental scenarios: training the model on the complete historical IHS dataset including the COVID-19 spike period and training the model on a cleaned dataset in which the extreme COVID-19 surge was removed. The evaluation uses MAE, RMSE, R^2 , MAPE, and Pearson Correlation, and the comparison is summarized in Table 1.

Table 1. Comparison of Model Evaluation Result on Full Data and Without COVID-19 Surge

Dataset	MAE	RMSE	R^2	MAPE	Pearson Correlation
Full Data	0.0125 = 0.01	0.02	0.95	1.31%	0.9770
Without Covid-19 Surge	0.0167 = 0.02	0.03	0.95	1.58%	0.9755

Based on Table 1, the model trained on the full dataset shows slightly better performance than the model trained on the cleaned dataset. The MAE on the full data is 0.0125 (approximately 1.25%), which is lower than 0.0167 (approximately 1.67%) on the dataset without the COVID-19 surge, and the RMSE is also lower on the full dataset (0.02 versus 0.03). The R^2 value indicates that the model explains 95% of the variance when trained on the full dataset, while it explains 93% of the variance in the cleaned-data scenario. The MAPE values remain low in both cases at 1.31% and 1.58%, showing that relative prediction errors are small under both conditions. Pearson correlation values are consistently high and close to 1 in both scenarios (0.9770 and 0.9755), indicating a strong linear agreement between predicted and actual values and suggesting that the model maintains robust behavior across both experimental settings. Overall, although including the COVID-19 spike slightly improves the numerical metrics, both configurations still demonstrate high predictive accuracy and stable performance.

4. Conclusions

The proposed CNN-BiGRU hybrid model with attention mechanism has been successfully applied for

predicting IHS stock prices using multivariate time series data. The experimental results under both full and cleaned datasets indicate that the model consistently achieves strong predictive accuracy, as evidenced by low MAE and RMSE values, high R^2 scores, low MAPE, and Pearson correlation coefficients close to 1. These metrics confirm the robustness and generalization capability of the model across volatile and stable market conditions.

Beyond numerical performance, the interpretability of the model was enhanced by the attention mechanism, which assigned higher weights to short-term moving averages and closing prices, reflecting the model's ability to emphasize meaningful patterns in financial data. This not only validates the predictive outcomes but also provides insights into which features and timesteps most significantly influence stock price trends.

Overall, the CNN-BiGRU with attention mechanism demonstrates strong potential as a reliable and interpretable forecasting tool for the Indonesian stock market. Its ability to capture complex temporal dynamics and highlight relevant features makes it valuable for supporting investment strategies, short-term trading decisions, and financial risk management. Future research can extend this work by integrating sentiment analysis from financial news or macroeconomic indicators to further enhance model robustness in highly dynamic market environments.

Acknowledgements

This research was supported by funding from the Directorate General of Higher Education, Research, and Technology, Ministry of Education, Culture, Research, and Technology of the Republic of Indonesia through the 2025 Regular Fundamental Research Grant Scheme. The authors express their sincere gratitude for the support provided.

References

- [1] W. Bao, Y. Cao, Y. Yang, H. Che, J. Huang, and S. Wen, "Data-driven stock forecasting models based on neural networks: A review," *Information Fusion*, vol. 113, pp. 1–20, Jan. 2025, doi: [10.1016/j.inffus.2024.102616](https://doi.org/10.1016/j.inffus.2024.102616).

- [2] G. Sun and Y. Li, "Intraday and Post-Market Investor Sentiment for Stock Price Prediction: A Deep Learning Framework with Explainability and Quantitative Trading Strategy," *Systems*, vol. 13, no. 5, pp. 1–24, May 2025, doi: [10.3390/systems13050390](https://doi.org/10.3390/systems13050390).
- [3] X. Kong et al., "Deep learning for time series forecasting: a survey," *International Journal of Machine Learning and Cybernetics*, vol. 16, no. 1, pp. 5079–5112, Aug. 2025, doi: [10.1007/s13042-025-02560-w](https://doi.org/10.1007/s13042-025-02560-w).
- [4] M. K. Anam, T. P. Lestari, L. Efrizoni, N. S. Handayani, and I. Andhika, "Sentiment Analysis Optimization Using Ensemble of Multiple SVM Kernel Functions," *Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi)*, vol. 9, no. 4, pp. 905–914, Aug. 2025, doi: [10.29207/resti.v9i4.6708](https://doi.org/10.29207/resti.v9i4.6708).
- [5] M. A. Lubis and Samsudin, "Using the Random Forest Method in Predicting Stock Price Movements," *Journal of Dinda Data Science, Information Technology, and Data Analytics*, vol. 5, no. 1, pp. 28–35, 2025, doi: [10.20895/dinda.v5i1.1765](https://doi.org/10.20895/dinda.v5i1.1765).
- [6] P. Chhajjer, M. Shah, and A. Kshirsagar, "The applications of artificial neural networks, support vector machines, and long-short term memory for stock market prediction," *Decision Analytics Journal*, vol. 2, p. 1, Mar. 2022, doi: [10.1016/j.dajour.2021.100015](https://doi.org/10.1016/j.dajour.2021.100015).
- [7] A. Rahmadeyan and Mustakim, "Long Short-Term Memory and Gated Recurrent Unit for Stock Price Prediction," in *Procedia Computer Science*, Elsevier B.V., 2024, pp. 204–212. doi: [10.1016/j.procs.2024.02.167](https://doi.org/10.1016/j.procs.2024.02.167).
- [8] S. Shinde, L. Wadhwa, N. Mohane, V. Pagar, N. Sherje, and S. Mane, "Stock Price Prediction using LSTM," in *2023 7th International Conference on Computing, Communication, Control And Automation, ICCUBEA 2023*, Institute of Electrical and Electronics Engineers Inc., 2023, pp. 1–7. doi: [10.1109/ICCUBEA58933.2023.10392023](https://doi.org/10.1109/ICCUBEA58933.2023.10392023).
- [9] G. Wojarnik, "The potential of convolutional neural networks for the analysis of stock charts," in *Procedia Computer Science*, Elsevier B.V., 2023, pp. 941–950. doi: [10.1016/j.procs.2023.10.081](https://doi.org/10.1016/j.procs.2023.10.081).
- [10] A. F. Azyus, S. K. Wijaya, and M. Naved, "Prediction of remaining useful life using the CNN-GRU network: A study on maintenance management," *Software Impacts*, vol. 17, pp. 1–6, Sep. 2023, doi: [10.1016/j.simpa.2023.100535](https://doi.org/10.1016/j.simpa.2023.100535).
- [11] Y. Lan, "A Hybrid CNN-LSTM Model for Stock Price Prediction with Spatial and Temporal Dependencies," in *CONF-FMCE 2025 Symposium: Semantic Communication for Media Compression and Transmission*, EWA Publishing, 2025, pp. 236–242. doi: [10.54254/2755-2721/155/2025.GL23570](https://doi.org/10.54254/2755-2721/155/2025.GL23570).
- [12] M. K. Anam, S. Defit, Haviluddin, L. Efrizoni, and M. B. Firdaus, "Early Stopping on CNN-LSTM Development to Improve Classification Performance," *Journal of Applied Data Sciences*, vol. 5, no. 3, pp. 1175–1188, 2024, doi: [10.47738/jads.v5i3.312](https://doi.org/10.47738/jads.v5i3.312).
- [13] M. K. Anam et al., "Improved Performance of Hybrid GRU-BiLSTM for Detection Emotion on Twitter Dataset," *Journal of Applied Data Sciences*, vol. 6, no. 1, pp. 354–365, Jan. 2025, doi: [10.47738/jads.v6i1.459](https://doi.org/10.47738/jads.v6i1.459).
- [14] F. M. P. Fozap, "Hybrid Machine Learning Models for Long-Term Stock Market Forecasting: Integrating Technical Indicators," *Journal of Risk and Financial Management*, vol. 18, no. 4, pp. 1–21, Apr. 2025, doi: [10.3390/jrfm18040201](https://doi.org/10.3390/jrfm18040201).
- [15] H. Liu, "A Hybrid CNN-LSTM Approach for Effective Stock Price Prediction in Optimizing Investment Strategies," in *Proceedings of the International Workshop on Navigating the Digital Business Frontier for Sustainable Financial Innovation (ICDEBA 2024)*, Atlantis Press, 2025, pp. 628–640. doi: [10.2991/978-94-6463-652-9_65](https://doi.org/10.2991/978-94-6463-652-9_65).
- [16] L. Xu, W. Xu, Q. Cui, M. Li, B. Luo, and Y. Tang, "Deep Heuristic Evolutionary Regression Model Based on the Fusion of BiGRU and BiLSTM," *Cognit Comput*, vol. 15, no. 5, pp. 1672–1686, Sep. 2023, doi: [10.1007/s12559-023-10135-6](https://doi.org/10.1007/s12559-023-10135-6).
- [17] M. K. Anam et al., "Sara Detection on Social Media Using Deep Learning Algorithm Development," *Journal of Applied Engineering and Technological Science*, vol. 6, no. 1, pp. 225–237, Dec. 2024, doi: [10.37385/jaets.v6i1.5390](https://doi.org/10.37385/jaets.v6i1.5390).
- [18] M. K. Anam, S. Defit, and R. Sovia, "Evaluating Word2Vec and FastText Embeddings in CNN-LSTM for Hotel Customer Satisfaction: Impact on Accuracy and Model Stability," in *International Conference on Data Science and Its Applications (ICoDSA)*, Jakarta: IEEE, 2025, pp. 540–545. doi: [10.1109/ICoDSA67155.2025.11157433](https://doi.org/10.1109/ICoDSA67155.2025.11157433).
- [19] G. Airlangga, "Stress Detection Using Hybrid Deep Learning Models with Attention Mechanisms: A Comparative Study of CNN-LSTM, CNN-GRU, and Ensemble Approaches," *Journal of Computer System and Informatics (JoSYC)*, vol. 6, no. 1, pp. 234–243, 2024, doi: [10.47065/josyc.v6i1.6284](https://doi.org/10.47065/josyc.v6i1.6284).
- [20] A. E. Minarno, M. Fadhlan, Y. Munarko, and D. R. Chandranegara, "Classification of Dermoscopic Images Using CNN-SVM," *JOIV: INTERNATIONAL JOURNAL ON INFORMATICS VISUALIZATION*, vol. 3, no. 2, pp. 606–612, 2024, doi: [10.62527/joiv.8.2.2153](https://doi.org/10.62527/joiv.8.2.2153).
- [21] L. Xu, W. Xu, Q. Cui, M. Li, B. Luo, and Y. Tang, "Deep Heuristic Evolutionary Regression Model Based on the Fusion of BiGRU and BiLSTM," *Cognit Comput*, vol. 15, no. 5, pp. 1672–1686, Sep. 2023, doi: [10.1007/s12559-023-10135-6](https://doi.org/10.1007/s12559-023-10135-6).
- [22] Q. Zhu, X. Jiang, and R. Ye, "Sentiment Analysis of Review Text Based on BiGRU-Attention and Hybrid CNN," *IEEE Access*, vol. 9, pp. 149077–149088, 2021, doi: [10.1109/ACCESS.2021.3118537](https://doi.org/10.1109/ACCESS.2021.3118537).
- [23] M. S. Islam, M. N. Islam, N. Hashim, M. Rashid, B. S. Bari, and F. Al Farid, "New Hybrid Deep Learning Approach Using BiGRU-BiLSTM and Multilayered Dilated CNN to Detect Arrhythmia," *IEEE Access*, vol. 10, pp. 58081–58096, 2022, doi: [10.1109/ACCESS.2022.3178710](https://doi.org/10.1109/ACCESS.2022.3178710).
- [24] Z. Hu, "Sentiment Analysis of Film Reviews Based on BiGRU+Attention+Capsule Fusion," *TechRxiv*, vol. 1, no. 1, pp. 1–8, 2021, doi: [10.36227/techrxiv.14863401.v1](https://doi.org/10.36227/techrxiv.14863401.v1).
- [25] B. A. S. Al-rimy, F. Saeed, M. Al-Sarem, A. M. Albarrak, and S. N. Qasem, "An Adaptive Early Stopping Technique for DenseNet169-Based Knee Osteoarthritis Detection Model," *Diagnostics*, vol. 13, no. 11, pp. 1–17, Jun. 2023, doi: [10.3390/diagnostics13111903](https://doi.org/10.3390/diagnostics13111903).
- [26] N. Wang and X. Zhao, "Time-Series Forecasting Based on Multi-Layer Attention Architecture," *KSII Transactions on Internet and Information Systems*, vol. 18, no. 1, pp. 1–14, Jan. 2024, doi: [10.3837/tis.2024.01.001](https://doi.org/10.3837/tis.2024.01.001).
- [27] L. Zhou, S. Yuan, Z. Feng, G. Dai, and G. Zhou, "A Lambda Layer-Based Convolutional Sequence Embedding Model for Click-Through Rate Prediction," *Wuhan University Journal of Natural Sciences*, vol. 29, no. 3, pp. 198–208, Jun. 2024, doi: [10.1051/wujns/2024293198](https://doi.org/10.1051/wujns/2024293198).
- [28] L. L. Van Fc, S. Defit, and S. Bukhori, "Monitoring and Reporting Unethical Actions in Higher Education Using the BiLSTM and BiGRU Algorithms," in *ICFTSS 2024 - International Conference on Future Technologies for Smart Society*, Institute of Electrical and Electronics Engineers Inc., 2024, pp. 69–74. doi: [10.1109/ICFTSS61109.2024.10690271](https://doi.org/10.1109/ICFTSS61109.2024.10690271).
- [29] A. Jierula, S. Wang, T. M. Oh, and P. Wang, "Study on accuracy metrics for evaluating the predictions of damage locations in deep piles using artificial neural networks with acoustic emission data," *Applied Sciences (Switzerland)*, vol. 11, no. 5, pp. 1–21, Mar. 2021, doi: [10.3390/app11052314](https://doi.org/10.3390/app11052314).
- [30] T. O. Hodson, "Root-mean-square error (RMSE) or mean absolute error (MAE): when to use them or not," *Geosci Model Dev*, vol. 15, no. 1, pp. 5481–5487, Jul. 2022, doi: [10.5194/gmd-15-5481-2022](https://doi.org/10.5194/gmd-15-5481-2022).
- [31] D. Chicco, M. J. Warrens, and G. Jurman, "The coefficient of determination R-squared is more informative than SMAPE, MAE, MAPE, MSE and RMSE in regression analysis evaluation," *PeerJ Comput Sci*, vol. 7, pp. 1–24, 2021, doi: [10.7717/PEERJ-CS.623](https://doi.org/10.7717/PEERJ-CS.623).

- [32] R. J. Dhanal and V. R. Ghorpade, "Aspect-based sentiment-analysis using topic modelling and machine-learning," *International Journal of Electrical and Computer Engineering*, vol. 14, no. 6, pp. 6689–6698, Dec. 2024, doi: [10.11591/ijece.v14i6.pp6689-6698](https://doi.org/10.11591/ijece.v14i6.pp6689-6698).
- [33] L. Muflikhah, T. N. Fatyanosa, N. Widodo, R. S. Perdana, Solimun, and H. Ratnawati, "Feature Selection for Hypertension Risk Prediction Using XGBoost on Single Nucleotide Polymorphism Data," *Healthc Inform Res*, vol. 31, no. 1, pp. 16–22, Jan. 2025, doi: [10.4258/hir.2025.31.1.16](https://doi.org/10.4258/hir.2025.31.1.16).
- [34] I. Made, D. C. Adyatma, and E. B. Setiawan, "Impact of Feature Selection on XGBoost Model with VGG16 Feature Extraction for Carbon Stock Estimation Using GEE and Drone Imagery," *Jurnal Ilmiah Teknik Elektro Komputer dan Informatika (JITEKI)*, vol. 10, no. 4, pp. 870–884, 2024, doi: [10.26555/jiteki.v10i4.30484](https://doi.org/10.26555/jiteki.v10i4.30484).
- [35] H. P. Ho, "The supplier selection problem of a manufacturing company using the weighted multi-choice goal programming and MINMAX multi-choice goal programming," *Appl Math Model*, vol. 75, pp. 819–836, 2019, doi: [10.1016/j.apm.2019.06.001](https://doi.org/10.1016/j.apm.2019.06.001).