Advance Sustainable Science, Engineering and Technology (ASSET)
Vol. 8, No.3, July 2026, pp. 0260301-01 ~ 0260301-010
ISSN: 2715-4211 DOI: https://doi.org/10.26877/asset.v8i3.2608

Comparative Deep Learning Models for Indonesian Gold Price Forecasting

Albi Pernata Jomantara Putra[1], Baginda Mi'raj Williamsyah[1,2], Achmad Rizal[1]*, Favian Dewanta[1], Anggunmeka Luhur Prasasti[1], Said Ziani[3]

[1]School of Electrical Engineering, Telkom University, Jl. Telekomunikasi No. 1, Bandung Terusan Buahbatu, Bandung, West Java, 40257, Indonesia

[2]Department of Electronics Engineering, Kookmin University, Seoul 02707, Republic of Korea

[3]Department of Health Technologies Engineering, Mohammed V University, Agdal 10106, Morocco

*achmadrizal@telkomuniversity.ac.id

Abstract. This study evaluates LSTM, CNN-LSTM, LSTM-GRU, and CNN-LSTM-GRU architectures for forecasting Indonesian gold prices using 1,269 daily observations (2022–2025). Models utilized Bayesian-optimized hyperparameters and were benchmarked against ARIMA-GARCH and Random Forest baselines across 30-day and 365-day horizons. Performance was assessed via MAE, RMSE, R^2, and MAPE, confirming deep learning's superiority in capturing non-linear dynamics over classical methods. The LSTM-GRU achieved the best numerical results, with MAPEs of 1.21% (short-term) and 1.32% (long-term). However, qualitative evaluation revealed that the highest-scoring model produced unstable long-term predictions, indicating a critical trade-off between numerical accuracy and forecast realism. These findings suggest financial model selection must prioritize stability alongside statistical metrics. A key limitation is the exclusive use of univariate data, necessitating future multivariate validation with macroeconomic indicators.

Keywords: Bayesian optimization, deep learning, gold price forecasting, LSTM-GRU, time-series prediction

(Received 2025-08-22, Revised 2026-02-01, Accepted 2026-03-06, Available Online by 2026-05-24)

1. Introduction

Gold has long served as both a commodity and a financial asset, valued for its rarity, durability, and role as a store of wealth [1]. In modern financial systems, it functions as a safe-haven asset, often appreciating during periods of economic uncertainty such as the 2008 global financial crisis and the COVID-19 pandemic. Despite its long-term stability, gold prices exhibit significant short-term volatility driven by interest rates, geopolitical tensions, currency movements, and market sentiment, making accurate forecasting both valuable and technically challenging.

A wide range of forecasting approaches has been proposed to model gold price dynamics. Deep learning methods—particularly Long Short-Term Memory (LSTM) networks—have demonstrated strong capability in capturing temporal dependencies in sequential data across both financial and applied AI domains [2]–[4]. Alongside these approaches, traditional statistical models such as ARIMA remain effective for modeling linear patterns but often struggle to represent the nonlinear behavior typical of commodity markets [5]–[7]. To address these limitations, machine learning methods such as Random Forest and Gradient Boosting have been increasingly adopted, showing improved predictive performance when multiple explanatory variables are available. Building on these developments, comparative studies involving CNN, LSTM, and GRU architectures highlight important trade-offs between predictive accuracy, robustness, and computational efficiency in time-series forecasting tasks [8]–[10].

Despite these advances, important gaps remain in literature. First, most gold price forecasting studies rely on multivariate inputs, leaving limited empirical evidence on how deep learning models perform using only historical price data. Second, many works focus on single architecture, making cross-model comparisons difficult. Most notably, previous studies rarely contrast LSTM-GRU against CNN-LSTM-GRU architectures under a univariate setting, particularly in emerging-market contexts. As a result, the accuracy–stability trade-offs of hybrid architectures for univariate financial forecasting remain insufficiently understood.

Addressing these gaps, this study conducts a systematic comparison of four deep learning architectures—LSTM, CNN-LSTM, LSTM-GRU, and CNN-LSTM-GRU—for forecasting daily Indonesian gold prices using a univariate dataset. Bayesian optimization is applied uniformly across models to ensure fair hyperparameter tuning. This study evaluates accuracy–stability trade-offs using real Indonesian market data, providing empirical insight into whether increasing architectural complexity yields more reliable forecasts.

The main objectives of this study are to: (1) evaluate the effectiveness of univariate deep learning models for gold price forecasting, (2) compare the performance of hybrid and non-hybrid architectures, and (3) assess the trade-off between predictive accuracy and forecast stability. The remainder of this paper is organized as follows: Section 2 describes the dataset and methodology; Section 3 presents experimental results and comparative analysis; and Section 4 concludes with implications and directions for future research.

2. Methods

This study employs a structured deep learning framework to forecast Indonesian gold prices using univariate time-series data. The methodology consists of data preparation and exploratory analysis, model construction, hyperparameter optimization, and quantitative performance evaluation.

2.1. Exploratory Data Analysis (EDA) and Pre-Processing

This study utilizes the Indonesian Gold Price Dataset obtained from the Bullion Rates website, covering the period from January 3, 2022, to June 24, 2025, with a total of 1,269 daily observations [11]. The dataset consists of two attributes: date and daily retail gold price in Indonesian Rupiah per gram. This univariate, high-frequency dataset was selected due to its relevance to consumer-level pricing behavior and its suitability for short- and long-term time-series forecasting.

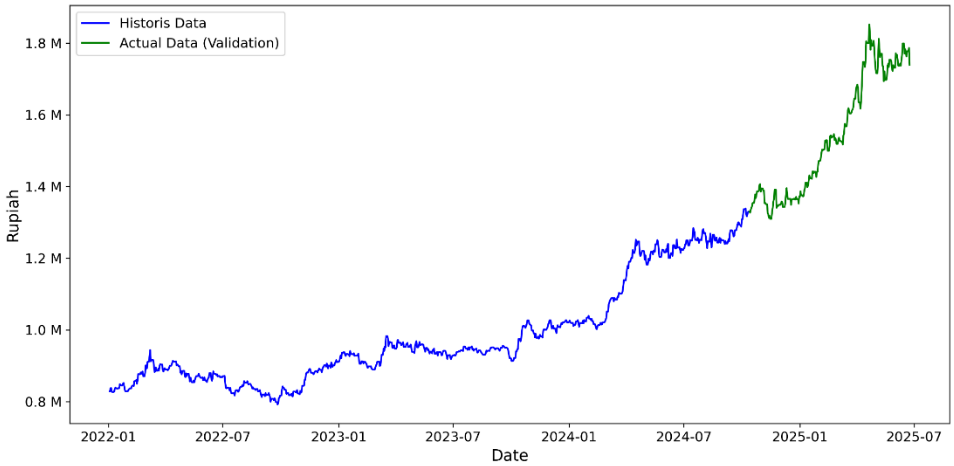

Figure 1. Indonesian gold price time series from January 2022 to June 2025 used for model training and evaluation.

An exploratory data analysis (EDA) was conducted to examine temporal dynamics and distributional properties. As illustrated in Figure 1, the gold price series exhibits clear non-stationary behavior, including sustained upward trends, abrupt price jumps, and volatility clustering during periods of economic stress. These characteristics motivate the use of nonlinear deep learning architectures, as linear statistical models such as ARIMA often underperform under such conditions [12].

During preprocessing, missing or inconsistent values were addressed using the forward-fill (FFill) imputation method, which preserves temporal continuity without introducing artificial trends [13]. The gold price series was then normalized to the range [0,1] using Min–Max scaling to improve numerical stability and facilitate neural network training.

The dataset was chronologically divided into 80% training data (1,015 observations) and 20% testing data (254 observations) to prevent information leakage and ensure evaluation on unseen future data. Within the training set, an internal validation split was applied during Bayesian hyperparameter optimization [14].

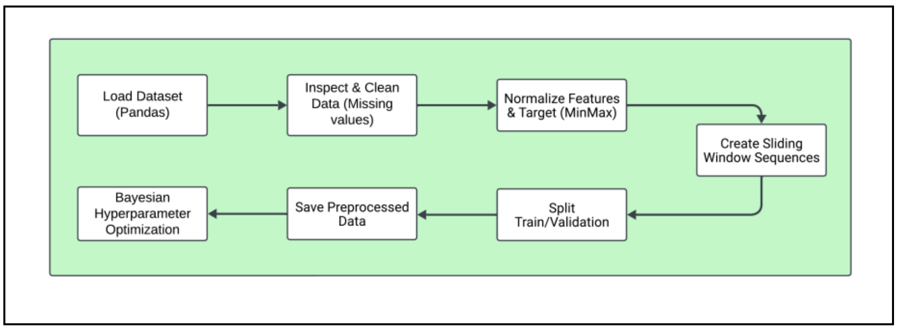

Figure 2. Data preprocessing workflow illustrates normalization, sliding-window sequence generation, and chronological data splitting.

Figure 2 illustrates the complete data preprocessing pipeline applied in this study. The raw gold price data were first imported from an Excel file using the Pandas library, followed by index resetting and data integrity checks. Missing values were handled using forward-fill imputation to preserve temporal continuity. The cleaned price series was then normalized using Min–Max scaling and transformed into supervised learning samples using a sliding-window approach. Finally, the processed sequences were reshaped into three-dimensional tensors and chronologically split into training, validation, and testing sets for model development and evaluation.

The dataset was chronologically split into 80% training (1,015 observations) and 20% testing (254 observations) to prevent information leakage [15]. A sliding-window technique was applied to generate input sequences for 30-day and 365-day horizons with an input shape of *(samples, timesteps, 1)*, where each sample consists of 30 consecutive price observations and one output corresponding to the next-day price [16]. Finally, these sequences were reshaped into 3D tensors to accommodate the convolutional and recurrent architectures [17]. The results of EDA and preprocessing directly informed the architectural design choices. LSTM layers were selected to capture long-term temporal dependencies, CNN layers were employed to extract short-term local patterns and suppress noise, and GRU layers were incorporated to improve convergence efficiency and reduce computational complexity. Previous studies have consistently reported that such hybrid deep learning architectures outperform conventional statistical methods in volatile commodity markets, particularly when modelling nonlinear temporal dynamics using univariate data [18].

2.2. Hybrid CNN-LSTM Model

The CNN–LSTM architecture utilizes a one-dimensional convolutional layer (Conv1D) as a front-end to extract local features and filter high-frequency noise. These feature maps are subsequently processed by an LSTM layer, which models long-range temporal dependencies. This design allows the LSTM to focus on structured temporal representations rather than raw price fluctuations, thereby improving generalization performance [19][20].

2.3. Hybrid LSTM-GRU Model

The LSTM–GRU model balances modeling power and computational efficiency by cascading an LSTM layer with a GRU layer. The LSTM captures complex long-term dependencies and passes the full sequence of hidden states to the GRU. The GRU layer then refines this temporal representation using fewer parameters, which enhances convergence speed and helps mitigate overfitting, particularly on moderate-sized datasets [21].

2.4. Hybrid CNN-LSTM-GRU Model

This hierarchical architecture integrates Conv1D, LSTM, and GRU layers to model nonlinear dynamics effectively. The Conv1D layer extracts salient local features, which are passed to an LSTM layer for long-term dependency modeling. A subsequent GRU layer refines the sequential representation to reduce computational overhead before a final Dense layer generates the prediction. This combination maximizes the strengths of feature extraction, long-range memory, and efficient sequential refinement [22]. Table 1 summarizes the functional roles and necessity of each deep learning component used in the proposed hybrid architecture:

Table 1. Functional Roles of CNN, LSTM, and GRU in the Proposed Forecasting Framework

Model	Primary Function	Temporal Role in This Study	Reason for Inclusion	Limitation Addressed by Hybridization
CNN (Conv1D)	Local pattern extraction	Captures short-term price fluctuations and local temporal structures	Enhances feature representation from raw price sequences	Lacks long-term dependency modeling
LSTM	Long-term dependency modeling	Learns long-range temporal relationships in gold price movements	Handles vanishing gradient issues and preserves long-term memory	Computationally intensive, slower convergence
GRU	Efficient temporal refinement	Refines sequential representations with fewer parameters	Improves training efficiency and generalization	Slightly weaker long-term memory than LSTM

2.5. Model Validation

Model performance was quantitatively evaluated using four established regression metrics.

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \widehat{y_i}| \tag{1}$$

The first evaluation metric used is the Mean Absolute Error (MAE), defined in equation (1), which measures the average absolute difference between the predicted values $\widehat{y_i}$ and the actual observations y_i, providing an intuitive estimate of error magnitude expressed directly in Rupiah.

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \widehat{y_i})^2} \tag{2}$$

The Root Mean Squared Error (RMSE), given in equation (2), computes the square root of the mean squared prediction error, thereby assigning greater penalty to large deviations and serving as an indicator of model stability under extreme forecasting errors.

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{y_i - \widehat{y_i}}{y_i}\right| \times 100\% \tag{3}$$

The Mean Absolute Percentage Error (MAPE), defined in equation (3), expresses the average prediction error as a percentage of the actual value, enabling scale-independent comparison of forecasting accuracy across models.

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(y_i - \overline{y_i})^2}{\sum_{i=1}^{n}(y_i - \hat{y})^2} \tag{4}$$

Finally, the coefficient of determination R^2, formulated in equation (4), quantifies the proportion of variance in the observed gold prices that is explained by the model predictions, with values closer to unity indicating stronger explanatory power. All evaluation metrics were computed using the sklearn.metrics library to ensure consistency, accuracy, and reproducibility of the experimental results [23].

2.6. Hyperparameter and Computational Environment

Bayesian Optimization was employed using the Keras Tuner framework to efficiently identify optimal hyperparameter configurations [24]. The search space included within Table 2:

Table 2. Hyperparameter Search Space

Parameter	Search Space
LSTM/GRU units	50–150
CNN filters	32–128
Kernel size	2–5
Dropout rate	0.2–0.4
Learning rate	1e−4–1e−2
Optimizer	Adam, RMSProp, Nadam, SGD

This strategy balances exploration and exploitation via a probabilistic surrogate model and acquisition function, enabling faster convergence than grid or random search [25]. Optimizer selection was treated as a tunable parameter to ensure architectural and learning-strategy compatibility [26]–[29]. All experiments were conducted using Python 3.x with TensorFlow/Keras, Pandas, NumPy, and Scikitlearn. Model training was performed on an ASUS ROG (2021) laptop equipped with an NVIDIA RTX 3050 GPU (4 GB VRAM) and an AMD Ryzen 5 5600H processor.

3. Results and Discussion

This section presents the experimental results of the four evaluated deep learning architectures like LSTM, CNN-LSTM, LSTM-GRU, and CNN-LSTM-GRU in forecasting daily gold prices in Indonesia. The models were trained on historical data from 2022 to 2025 and tested across short-term (30-day) and long-term (365-day) horizons. Model performance is evaluated using MAE, RMSE, R^2, and MAPE, followed by statistical significance testing and qualitative trend analysis. The discussion emphasizes accuracy, stability, and generalization capability, with findings contextualized against existing literature.

3.1. Model Parameters

The final hyperparameter configurations obtained via Bayesian Optimization for both forecasting horizons are summarized in Table 3.

Table 3. Hyperparameter Tuning Model within 30 Days and 365 Days

Parameter	LSTM		LSTM-GRU		CNN-LSTM		CNN-LSTM-GRU	
	30 days	**365 days**	**30 days**	**365 days**	**30 days**	**365 days**	**30 days**	**365 days**
Epoch	100	100	100	100	100	100	100	100
Batch Size	16	16	16	16	16	16	16	16
Units	100,150	100,100	150,100	150,150	150	150	150,150	50,150
Dropout	0.5;0.4	0.6;0.6	0.4;0.2	0.2;0.2	0.4;0.2	0.5;0.3	0.2;0.4;0.3	0.2;0.3;0.4
Filter (Kernel Size)	-	-	-	-	128(2)	128(3)	64(3)	32(2)
Optimizer	Adam	Adam	Adam	Nadam	Adam	Adam	Adam	Adam

All models converged within 100 epochs using a batch size of 16, ensuring consistent training conditions. While hybrid architectures employed convolutional filters and multiple recurrent layers, simpler configurations consistently emerged as optimal, particularly for univariate long-horizon forecasting. This early indication suggests diminishing returns from excessive architectural depth when input information is limited.

3.2. Actual vs Predicted Price Simulation

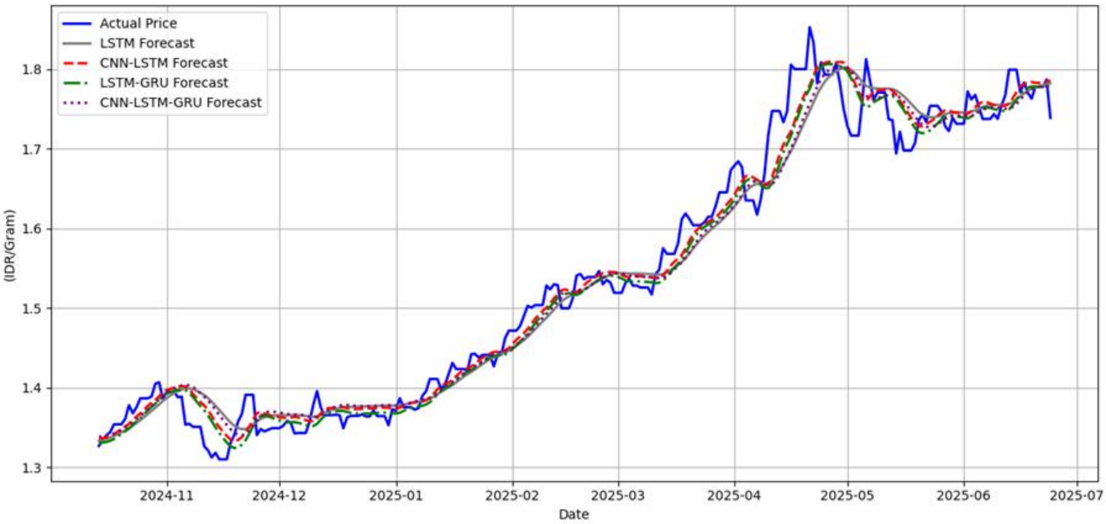

Figure 3. Actual vs Predicted Price Illustration

Figure 3 compares actual gold prices (IDR/gram) against predictions from the four models between late 2024 and mid-2025. While all models captured the general upward trend and the March 2025 rally, notable differences in responsiveness emerged. The LSTM-GRU and CNN-LSTM-GRU models demonstrated superior adaptability, closely tracking rapid surges and corrections, particularly during the volatile phase in April 2025. Conversely, the CNN-LSTM model produced a smoother curve that effectively filtered noise but underestimated peaks, while the standalone LSTM exhibited lag during sudden fluctuations. These results highlight the advantage of hybrid architecture in balancing feature extraction with responsiveness.

3.3. Model Performance Comparison

Table 4 summarizes the forecasting performance of all evaluated models, including deep learning architecture and classical baselines, under both short-term (30-day) and long-term (365-day) horizons. Across both horizons, the LSTM–GRU model consistently achieved the best overall performance, yielding the lowest error metrics and highest explanatory power. For the 30-day horizon, LSTM–GRU recorded a MAPE of 1.21%, RMSE of 25,435, and an R^2 of 0.976, indicating highly accurate short-term predictions. The CNN–LSTM model followed closely, while the more complex CNN–LSTM–GRU architecture underperformed relative to simpler hybrids, suggesting diminishing returns from increased architectural depth and a higher susceptibility to overfitting in short-horizon settings. In contrast, classical baselines exhibited substantially weaker performance, with ARIMA–GARCH and Random Forest producing negative R^2 values (–0.41 and –1.38, respectively), reflecting their inability to capture nonlinear temporal dependencies when limited to univariate inputs.

A general degradation in forecasting accuracy was observed when extending the prediction horizon to 365 days, which is expected due to the accumulation of uncertainty in long-range forecasts. Despite this, the LSTM–GRU model remained the most robust, achieving a MAPE of 1.32% and an R^2 of 0.972, demonstrating strong capability in modeling long-term temporal dependencies using only historical price information. The CNN–LSTM–GRU model achieved moderate accuracy (MAPE = 1.50%, R^2 = 0.965) but again failed to outperform the simpler LSTM–GRU configuration, reinforcing the conclusion that increased model complexity does not necessarily translate into superior generalization in univariate forecasting contexts.

Table 4. Model Evaluation of National Gold Prediction within 30 Days and 365 Days

Model	**MAE**		**RMSE**		**R^2**		**MAPE**	
	30 days	**365 days**	**30 days**	**365 days**	**30 days**	**365 days**	**30 days**	**365 days**
LSTM	23,964	26,157	31,722	34,476	0.963	0.957	1.53%	1.71%
CNN-LSTM	19,577	24,605	25,843	30,856	0.976	0.965	1.25%	1.61%
LSTM-GRU	19,104	20,844	25,435	27,634	0.976	0.972	1.21%	1.32%
CNN-LSTM-GRU	22,976	23,294	30,727	30,839	0.966	0.965	1.46%	1.50%
ARIMA-GARCH	150,195	156,721	198,464	203,232	-0.41	-0.51	8.97%	10.21%
Random Forest	197,050	199,352	256,406	276,874	-1.38	-1.64	11.77%	14.43%

Overall, these results highlight the effectiveness of gated recurrent hybrids for both short- and long-term gold price forecasting, while underscoring the limitations of classical statistical and non-sequential machine learning models under nonlinear and volatile market conditions.

3.4. Model Trend Analysis

Figures 4–5 present representative forecast trajectories for both short- and long-term horizons and are discussed jointly to avoid redundancy. Across all models, distinct differences emerge in how trend direction, volatility, and stability are handled. The LSTM and CNN–LSTM architectures produce relatively smooth and monotonic forecasts, indicating a strong bias toward dominant trend extrapolation.

While this behavior contributes to stable predictions, it also reflects limited responsiveness to short-term volatility and sudden market fluctuations, which are characteristic of gold price dynamics.

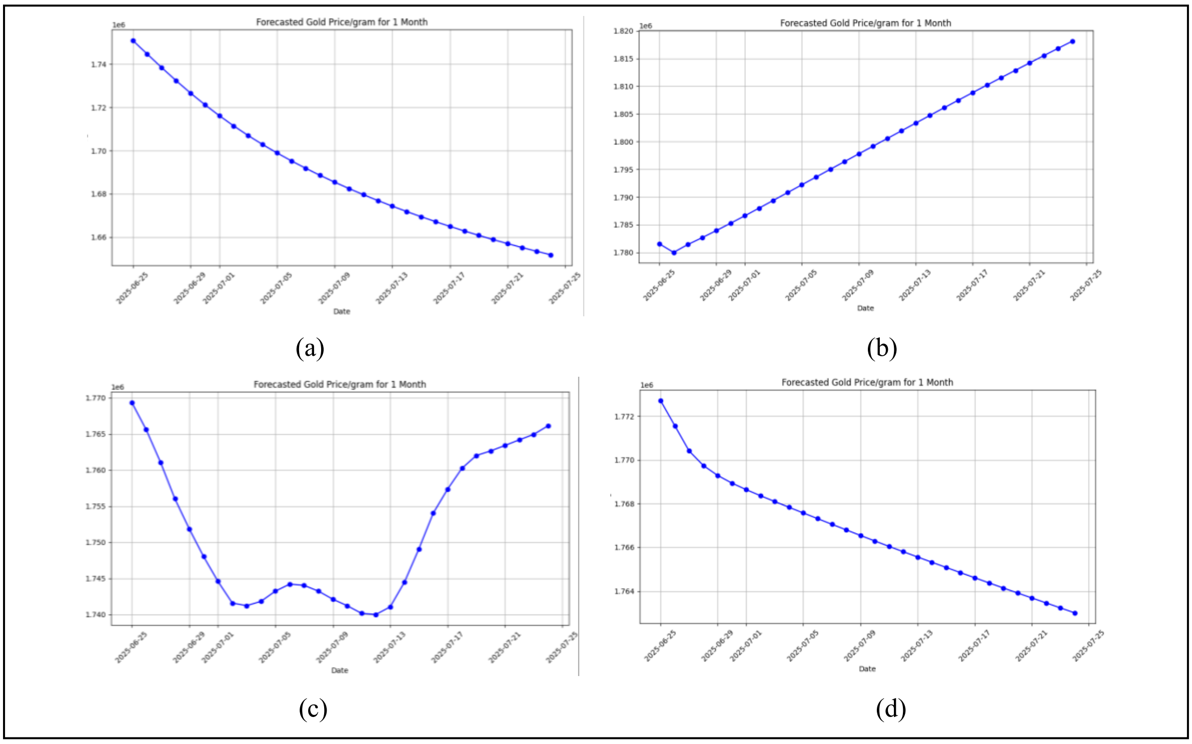

Figure 4. Model Trend Prediction for one month: (a) LSTM, (b) CNN-LSTM, (c) LSTM-GRU, (d) CNN-LSTM-GRU

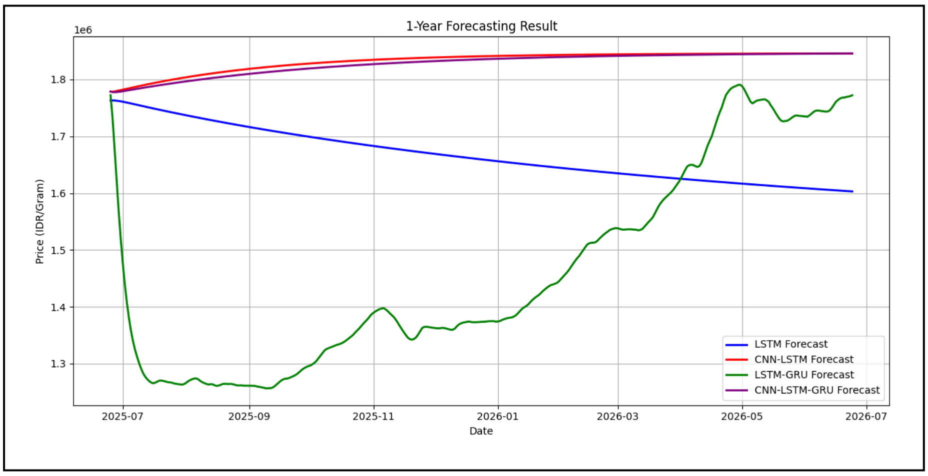

Figure 5. Model Trend Prediction for one year

The LSTM–GRU model exhibits high dynamism, achieving superior metrics but generating unrealistic long-term deviations, which suggests overfitting to localized patterns. Conversely, the CNN–LSTM–GRU model demonstrates smoother trajectories due to convolutional noise filtering, yet this stability results in delayed responses to abrupt price changes. This highlights a critical accuracy–stability trade-off in univariate settings, consistent with Malik et al. [22] regarding the potential for overfitting and instability in GRU-enhanced hybrids.

3.5. *Discussion and Literature Alignment*

Experimental results indicate that increasing architectural complexity does not guarantee superior univariate forecasting performance; simpler hybrids (LSTM–GRU, CNN–LSTM) demonstrated better generalization than the deeper CNN–LSTM–GRU. This aligns with Malik et al. [22], who cautioned against overfitting and instability in deep univariate models. Although all models achieved robust metrics ($R^2 > 0.95, MAPE < 2\%$), qualitative analysis highlighted that numerical accuracy does not imply forecast realism. Thus, moderately complex architectures are preferred to balance predictive accuracy with the stability required for practical financial decision-making.

3.6. *Statistical Validation of Model Superiority*

Table 5 reports the results of paired statistical tests used to validate performance differences among forecasting models over the 30-day and 365-day horizons.

Table 5. Statistical Validation of Model Evaluation of National Gold Prediction

Days	Comparison	Test	p-value	Effect Size	Magnitude
30	LSTM–GRU vs LSTM	Wilcoxon	0.003	r = 0.71	Large
		Paired t-test	0.005	d = 1.12	Large
	LSTM–GRU vs CNN–LSTM	Wilcoxon	0.018	r = 0.54	Moderate
		Paired t-test	0.021	d = 0.78	Moderate
	LSTM–GRU vs CNN–LSTM–GRU	Wilcoxon	0.062	r = 0.31	Small (marginal)
365	LSTM–GRU vs LSTM	Wilcoxon	< 0.001	r = 0.76	Large
		Paired t-test	< 0.001	d = 1.34	Very large
	LSTM–GRU vs CNN–LSTM	Wilcoxon	0.009	r = 0.59	Moderate–large
		Paired t-test	0.011	d = 0.86	Large
	LSTM–GRU vs CNN–LSTM–GRU	Wilcoxon	0.021	r = 0.48	Moderate

Performance differences were validated using Wilcoxon signed-rank and paired t-tests on rolling-window errors to account for non-normality. For the 30-day horizon, the LSTM–GRU significantly outperformed LSTM ($p = 0.003, r = 0.71$) and CNN–LSTM ($p = 0.018, r = 0.54$), though gains against the deeper CNN–LSTM–GRU were marginal ($p = 0.062$), indicating diminishing returns from complexity. In the 365-day horizon, LSTM–GRU's superiority became more pronounced, showing significant improvements over all architectures, including CNN–LSTM–GRU ($p = 0.021$). Furthermore, all deep learning models demonstrated statistically, and practically, significant dominance over classical baselines ($p < 0.001, r > 0.85$).

4. **Conclusion**

This study evaluated Bayesian-optimized deep learning models for forecasting Indonesian gold prices across 30-day and 365-day horizons. The LSTM–GRU architecture achieved the lowest validation errors (MAPE 1.21%–1.32%) but exhibited unrealistic long-term volatility, whereas the CNN–LSTM model provided more stable and plausible forecasts despite marginally higher errors. These findings highlight that numerical accuracy alone is insufficient for practical investment systems, such as daily investment decision support tools or algorithmic trading frameworks; forecast stability must be prioritized. Future research should address the limitations of this univariate approach by incorporating multivariate macroeconomic indicators to enhance model robustness.

Declaration of AI and AI assisted technologies in the writing process
During the preparation of this work the author(s) used PaperPal in order to check and improve the grammar. After using this tool/service, the author(s) reviewed and edited the content as needed and took full responsibility for the content of the publication

Declaration of Competing Interest
The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements
The authors would like to express their sincere gratitude to Bullion Rates for providing valuable market data that formed the basis of this research. Appreciation is also extended to Telkom University for the academic support, research facilities, and resources that enabled the successful completion of this study. Their contributions have been instrumental in ensuring the quality and relevance of the research outcomes.

References
[1] H. D. Fadly and F. Arifin, "Indonesian Gold Price Prediction: A Machine Learning Approach Using Random Forest Regressor," Int. J. Comput. Sci. Mobile Comput., vol. 14, no. 2, pp. 32–43, Feb. 2025. DOI: https://doi.org/10.47760/ijcsmc.2025.v14i02.004
[2] M. Mokashi et al., "A Comparative Analysis of Long Short-Term Memory Networks and Artificial Neural Networks for Gold Price Prediction," J. Inf. Syst. Eng. Manag., vol. 10, no. 34s, 2025.
[3] N. K. Saikumar et al., “Enhanced air quality prediction using AI: A comparative study of GRU, CNN, and XGBoost models,” Adv. Sustain. Sci. Eng. Technol., vol. 7, no. 3, Art. no. 02503012, Jun. 2025. DOI: https://doi.org/10.26877/asset.v7i3.1589
[4] I. D. Raharjo and N. E. R. Subhiyakto, “Implementing long short-term memory (LSTM) in chatbots for Multi Usaha Raya,” Adv. Sustain. Sci. Eng. Technol., vol. 6, no. 4, Art. no. 02404018, Oct. 2024. DOI: https://doi.org/10.26877/asset.v6i4.934
[5] L. Gasper and H. Mbwambo, “Forecasting Crude Oil Prices by Using ARIMA Model: Evidence from Tanzania,” Journal of Accounting, Finance and Auditing Studies, vol 9, no. 2, 2023. DOI: https://doi.org/10.32602/jafas.2023.017
[6] L. K. Shrivastav and R. Kumar, “An ensemble of random forest gradient boosting machine and deep learning methods for stock price prediction”, Journal of Information Technology Research (JITR), vol. 15 no. 1. 2022. DOI: https://doi.org/10.4018/JITR.2022010102
[7] S. Setyowibowo, M. As’ad, S. Sujito, and E. Farida, “Forecasting of Daily Gold Price Using ARIMA–GARCH Hybrid Model,” J. Ekonomi Pembangunan, vol. 19, no. 2, pp. 257–270, 2022. DOI: https://doi.org/10.29259/jep.v19i2.13903
[8] Y. Luo, "A Research of Fine-Tuning CNN-LSTM Model for Gold Price Prediction," Proc. Finance in the Age of Environmental Risks and Sustainability - ICFTBA, 2024. DOI: https://doi.org/10.54254/2754-1169/94/2024ox0185
[9] Y. Guo, " Research on the application of gold price prediction based on LSTM mode," Information Systems and Economics, vol. 5 no.4, 2024. DOI: 10.23977/infse.2024.050414
[10] Z. Wang, J. Yan, and H. O. Wang, “Deep learning for time series forecasting: A survey,” IEEE Transactions on Neural Networks and Learning Systems, vol. 32, no. 11, pp. 4761–4783, 2021. DOI: https://doi.org/10.1109/ACCESS.2024.3422528
[11] “Current gold prices in Indonesian rupiahs (IDR).” Bullion-Rates.com, [Online]. Available: https://www.bullion-rates.com. [Accessed: Jun. 2025].
[12] S. Livieris, G. Pintelas, and P. Pintelas, “A CNN–LSTM model for gold price time-series forecasting,” Neural Computing and Applications, vol. 33, pp. 13559–13568, 2021. DOI: https://doi.org/10.1007/s00521-020-04867-x

[13] R. J. Hyndman and G. Athanasopoulos, Forecasting: Principles and Practice, 3rd ed. Melbourne, Australia: OTexts, 2021. [Online]. Available: https://otexts.com/fpp3/
[14] A. Amini and M. Kalantari, "Gold price prediction by a CNN-Bi-LSTM model along with automatic parameter tuning," PlosOne, Mar 7;19(3):e0298426, 2024. DOI: https://doi.org/10.1371/journal.pone.0298426
[15] H. Fischer and B. Krauss, "Deep learning with long short-term memory networks for financial market predictions," European Journal of Operational Research, vol. 270, no. 2, pp. 654–669, 2018. DOI: https://doi.org/10.1016/j.ejor.2017.11.054
[16] Y. Zhang, M. Liu, and S. X. Yang, "Time series forecasting using a hybrid ARIMA and neural network model," Neurocomputing, vol. 50, pp. 159–175, 2003. DOI: https://doi.org/10.1016/S0925-2312(01)00702-0
[17] J. Brownlee, "How to Convert a Time Series to a Supervised Learning Problem in Python," Machine Learning Mastery, [Online]. Available: https://machinelearningmastery.com/convert-time-series-supervised-learning-problem-python/ . [Accessed: Jul. 2, 2025].
[18] Y. Kim and H. Kim, "Forecasting stock prices with a feature fusion LSTM-CNN model using different representations of the same data," PLOS ONE, vol. 16, no. 6, p. e0252313, 2021. https://doi.org/10.1371/journal.pone.0212320
[19] I. W. K. G, Santika, S. Sa'adah, and P. E. Yunanto, "Gold price prediction using convolutional neural network-long short-term memory (CNN-LSTM)", Kinetik: Game Technology, Information System, Computer Network, Computing, Electronics, and Control, vol. 6 no. 3, 2021.DOI: https://doi.org/10.22219/kinetik.v6i3.1253
[20] N. Rouf, M. B. Malik, T. Arif, S. Sharma, S. Singh, S. Aich, and H. C. Kim, "Stock market prediction using machine learning techniques: a decade survey on methodologies, recent developments, and future directions", Electronics, vol.10 no. 21, 2717, 2021. DOI: https://doi.org/10.3390/electronics10212717
[21] G. R. Patra and M. N. Mohanty," An LSTM-GRU based hybrid framework for secured stock price prediction", Journal of Statistics and Management Systems, vol. 25 no. 6, 2022. DOI: https://doi.org/10.1080/09720510.2022.2092263
[22] W. Fu," Financial Time Series Data Forecasting based on CEEMDAN-BiGRU-BO Model", Frontiers in Economics and Management, vol. 6 no. 12, 90-105. 2025. DOI: https://doi.org/10.6981/FEM.202512_6(12).0008
[23] G. Taneva-Angelova, S. Raychev, and G. Ilieva, "A framework for gold price prediction combining classical and intelligent methods with financial, economic, and sentiment data fusion," International Journal of Financial Studies, vol. 13, no. 2, p. 102, 2025. DOI: https://doi.org/10.3390/ijfs13020102
[24] G. S., Vidya and V. S.Hari, "Gold price prediction and modelling using deep learning techniques", In 2020 IEEE Recent Advances in Intelligent Computational Systems (RAICS) (pp. 28-31). IEEE. 2020. DOI: https://doi.org/10.1109/RAICS51191.2020.9332471
[25] J. Bergstra, R. Bardenet, Y. Bengio, and B. Kégl, "Algorithms for hyper-parameter optimization," in Advances in Neural Information Processing Systems, vol. 24, 2011.
[26] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," in Proc. of the 3rd International Conference on Learning Representations (ICLR), 2015. [Online]. Available: https://arxiv.org/abs/1412.6980
[27] T. Dozat, "Incorporating Nesterov Momentum into Adam," ICLR Workshop, 2016. [Online]. Available: https://openreview.net/forum?id=OM0jvwB8jIp57ZJjtNEZ
[28] T. Tieleman and G. Hinton, "Lecture 6.5 - RMSProp: Divide the gradient by a running average of its recent magnitude," COURSERA: Neural Networks for Machine Learning, 2012.
[29] L. Bottou, "Large-scale machine learning with stochastic gradient descent," in Proceedings of COMPSTAT, Heidelberg: Physica-Verlag HD, pp. 177–186. 2010. DOI: https://doi.org/10.1007/978-3-7908-2604-3_16