Vol. 4 No. 2, July 2026 (68-77)
ISSN: 2987-6990
DOI: https://doi.org/10.60005/coreid.v4i2.182

Analysis of Feature Engineering on LSTM and GRU Forecasting Performance Across Food Commodities with Different Volatility Levels

Zalid Qomalita Hijriana[1], Muhammad Mulyawan[2], Lina Alfaridah ZH[3], Widya Puteri Aulia[4]
[1,3,4]Department of Informatics, UIN Sunan Gunung Djati Bandung
[2]Data and Statistics Division, Bandung City Communication and Informatics Agency

Article Info

Article history:

Received June 18, 2026
Revised August 06, 2026
Accepted August 07, 2026

Keywords:

Deep learning
Food commodity forecasting
Price volatility
Temporal feature engineering
Time series forecasting

ABSTRACT

Food commodity price prediction plays a crucial role in monitoring food price stability and inflation, particularly for commodities with varying volatility characteristics. This study compares the performance of Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) models in forecasting daily food commodity prices in Bandung City, using chili and rice dataset. Daily price data were collected and modeled in two experimental scenarios: forecasting using only historical price data and forecasting with temporal feature engineering. The results showed that GRU consistently outperformed LSTM across all experiments. Without feature engineering, GRU achieved a MAPE score 0.49% and 3.26% for rice and chili, respectively. The incorporation of temporal features improved forecasting performance, reducing forecasting errors by up to 31.30% for rice and 29.55% for chili. The best overall performance was achieved by the GRU model with temporal feature engineering (MAPE rice = 0.48% and MAPE chili = 2.48%). These findings indicate that incorporating feature engineering remains beneficial for enhancing deep learning models performance. This study contributes to the development of deep learning methods for food commodity price forecasting and provides an understanding of the role of temporal feature engineering in handling commodities with different levels of volatility.

Corresponding Author:

Zalid Qomalita Hijriana
Department of Informatics, Faculty of Science and Technology, UIN Sunan Gunung Djati Bandung
Jl. A. H. Nasution No. 105, Cibiru, Bandung, Indonesia. 40614
Email: zalidqomalita@uinsgd.ac.id

1. INTRODUCTION

Food commodity prices are one of the important factors affecting economic stability and inflation level in a region [1], [2]. This food price stability has also become a strategic issue in Bandung City as fluctuations in essential commodity prices can directly affect household purchasing power and regional inflation. The Bandung City Government has intensified efforts to monitor food prices and market supply conditions to anticipate potential price increases and maintain economic stability. In this regard, accurate commodity price forecasts can provide significant benefits to various stakeholders. For consumers, price forecasts offer early information on potential future price movements, while for policymakers, forecasts can serve as an early warning system to support market monitoring, inflation control, and evidence based decision making.

Some commodities such as chilies and rice often receive attention due to their significant contribution to food inflation, particularly within the volatile food category [3], [4], [5]. Fluctuations in food commodity prices can be influenced by various factors, such as weather conditions, supply distribution, market demand, harvest seasons, and religious holidays. These food commodities present different volatility characteristics. Chili is categorized as a highly volatile commodity because its price is sensitive to supply changes and distribution conditions. In contrast, rice tends to have more stable prices since it is a staple commodity that receives special attention in price control and distribution management. These differences of volatility level present a unique challenge and indicate that forecasting approaches should consider the temporal patterns of each commodity.

Several forecasting food commodity prices approaches have been developed to support decision-making and an early warning system for potential food price increases. Traditional statistical methods like the Autoregressive Integrated Moving Average (ARIMA) have been widely used in time series forecasting. However, these methods have limitations in capturing nonlinear patterns and long term dependencies in dynamic and complex daily price data [6], [7].

The development of deep learning has provided a better alternative for modeling time series data. Recurrent Neural Network (RNN) models, particularly Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU), are capable of learning temporal patterns and long term dependencies from sequential data. LSTM utilizes a memory cell mechanism to retain information over long periods [8], [9], [10]. GRU offers a simpler architecture with fewer parameters than LSTM, resulting in more efficient training processes [11], [12], [13]. Both models have been widely applied in various forecasting tasks, including commodity price and financial market forecasting. In addition, model architecture forecasting performance can also be influenced by the feature engineering process [14], [15]. In time series data, temporal feature engineering techniques such as lag features, rolling mean, and price difference features can help models better understand historical patterns and price movement trends. Although deep learning models such as LSTM and GRU are designed to automatically learn temporal patterns, there remains ongoing discussion whether feature engineering is still necessary to improve forecasting performance especially for commodity price prediction.

Based on these issues, this study aims to compare the performance of LSTM and GRU models in forecasting daily food commodity prices in Bandung City using two commodities with different volatility characteristics, namely chili and rice. This study also analyzes the impact of temporal feature engineering on forecasting performance for each model and commodity. The main contribution of this study is to provide an analysis of the effectiveness of temporal feature engineering in deep learning models for commodities with different volatility levels, as well as to compare the performance stability between LSTM and GRU models in daily food price forecasting. The results of this study are expected to support the development of food price monitoring systems and provide references for future research in the fields of food commodity forecasting and food inflation monitoring.

2. METHOD

This study adopts the Cross Industry Standard Process for Data Mining (CRISP-DM) methodology as the primary framework for developing the food commodity price forecasting model. CRISP-DM is widely used in data mining and machine learning research because it provides a systematic and structured approach for solving data-driven problems. This methodology consists of several stages executed in iteration such as business understanding, data understanding, data preparation, modeling, evaluation, and deployment as seen in Figure 1.

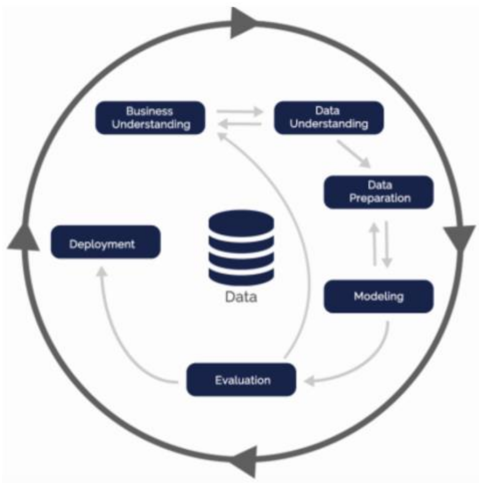

Figure 1. CRISP-DM Methodology [16]

2.1. Business Understanding

This stage focuses on identifying the research objectives. This study focused on the forecasting problems related to food commodity prices. Price fluctuations in food commodities, particularly highly volatile commodities such as chili, can significantly affect food inflation and market stability. Therefore, accurate forecasting methods are needed to support early monitoring of food price movements. This study aims to compare the forecasting performance of LSTM and GRU models for daily food commodity price forecasting. In addition, this study investigates the impact of temporal feature engineering on commodities with different volatility characteristics, namely chili as a highly volatile commodity and rice as a relatively stable commodity. The input of this study is daily price data in the form of a time series and the output is the prediction of the next price. This univariate setting was selected to ensure that the forecasting performance primarily reflects the temporal dynamics and volatility characteristics of the commodity price series rather than the influence of external factors.

2.2. Data Understanding

In the data understanding stage, the data collection, exploration, and initial analysis of the dataset used in the research were conducted. The dataset used in this study consists of daily food commodity prices in Bandung City, obtained from the Strategic Food Price Information Center (PIHPS). PIHPS is an official platform from Bank Indonesia that provides food price information from various regions in Indonesia and is often used as a reference for monitoring food price stability and inflation.

This study used two types of food commodities with different volatility characteristics: chili peppers which exhibit high volatility, and rice which exhibits low volatility. Data were collected from January 2021 to February 2026 at a daily frequency. Figure 2 presents a representative sample of the daily price movements to illustrate the contrasting volatility characteristics of chili and rice prices. Each data point consists of date and commodity price attributes. Supporting the volatility analysis, this study also calculated descriptive statistics such as average price, standard deviation, and coefficient of variation (CV) as shown in Table 1. Higher standard deviation and CV values in the chili data indicate that this commodity has a higher level of volatility than rice.

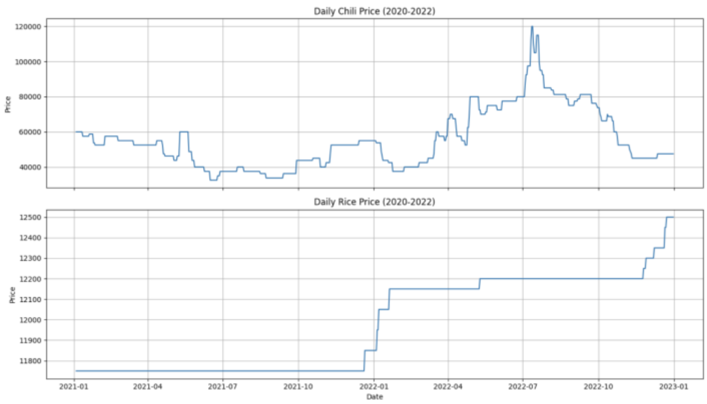

Figure 2. Sample daily price movements of chili and rice commodities in Bandung City, illustrating the higher volatility of chili prices compared to rice prices

Table 1. Descriptive Statistics of Commodity Prices

Commodity	Average Price (IDR)	Standard Deviation (IDR)	Coefficient of Variation (CV) (%)
Chili	60639.04	16488.23	27.19
Rice	13601.62	1408.96	10.35

2.3. Data Preparation

In the data preparation stage, a series of preprocessing processes are carried out to prepare the dataset before it is used in deep learning model training. This stage aims to ensure data quality, maintain temporal consistency and continuity, and create a suitable data structure for forecasting models. In this study, the raw data (as seen in Figure 3) still contains missing observations. These missing observations appear because there is no price recording during the holidays. Since this study needed complete daily price data, the missing values were handled using the forward fill method which propagates the last valid observation forward to the next temporary missing value. It assumes that the price remains unchanged during the holiday. The date attributes also need to be converted to a datetime format to facilitate time series data manipulation. The dataset is sorted chronologically by date to maintain temporal order in the forecasting process.

	date	chili_price
0	2021-01-04	60000
1	2021-01-05	60000
2	2021-01-06	60000
3	2021-01-07	60000
4	2021-01-08	60000
5	2021-01-11	60000
6	2021-01-12	57500
7	2021-01-13	57500
8	2021-01-14	57500
9	2021-01-15	57500

Figure 3. Raw data still contains missing observations that can disrupt the temporal structure required for time series forecasting.

This study uses a univariate approach with historical prices as the primary variable. Furthermore, temporal feature engineering is performed to generate additional features that can help the model better understand price change patterns. Some of the temporal features used include: (1) lag features, which are prices from previous periods [17]; (2) rolling mean, which is the moving average of prices over a specific period to capture local trends[18][19]; (3) price difference between periods, represents the dynamics of price increases and decreases. Table 2 summarizes the generated temporal feature formula for this study.

Table 2. Formula for Temporal Features

Features	Formula	Definition
Lag features	$Lag_{t-k} = P_{t-k}$	P_t is price in time t, k is lag period (set with lag-1 and lag-7).
Rolling mean	$RM_t = \frac{1}{n}\sum_{i=0}^{n-1} P_{t-i}$	RM_t is rolling mean in time t, n is windows size (set the window size = 7)
Price difference	$\triangle P = P_t - P_{t-1}$	P_t is current price then P_{t-1} is previous price

The next stage is data normalization using the standard scaler method [20]. Under this approach, each feature value is standardized by subtracting the corresponding feature mean and dividing by its standard deviation as written in (1). This process reduces scale differences among variables and generates features with a mean of zero and a variance of one. This normalization makes the deep learning model training process more stable and converging faster.

$$x'_i = \frac{x_i - x}{s} \tag{1}$$

where x_i is the original value, $\underline{x}$ is the mean of feature, s is the standard deviation of the feature, and x' is the standardized value.

The scaler fitting process is only performed on the training data (training set) to avoid data leakage, and then the same scaler is applied to the validation and testing data.The dataset is then divided into three parts: the training data (training set), the validation data (validation set), and the testing data (testing set), using a chronological partitioning with a ratio of 70:15:15. This partitioning is performed without shuffling to preserve the temporal characteristics of time series data.

Next, the data is converted into sequence form using a sliding window approach. In this process, historical data from several previous periods is used as input to predict prices in the next period. The sequence length is determined based on experimental results to capture temporal dependencies in daily price data. The resulting data are then used as inputs for training and evaluating the LSTM and GRU forecasting models.

2.4. Modeling

In this stage, a deep learning model was developed and trained to forecast daily food commodity prices. This study used two Recurrent Neural Network (RNN) architectures: LSTM and GRU. Both models were chosen because of their ability to learn temporal patterns and long term dependencies in time series data.

2.4.1 LSTM

LSTM is a development of the RNN architecture designed to overcome the vanishing gradient problem in sequential data modeling [21]. LSTM has been widely used in various time series forecasting tasks due to its ability to learn long term dependencies. Unlike conventional RNN, LSTM contains a memory cell mechanism that enables the model to store and preserve important information from previous time steps over longer periods, as seen in Figure 4. This structure allows LSTM to effectively capture temporal patterns and relationships among historical data during the forecasting process. The LSTM architecture consists of a cell state and three main gates: forget gates, input gates, and output gates [22]. The forget gate determines which information from the previous time step should be retained or removed from the cell state. This mechanism helps the model discard irrelevant information during prediction. The formula for forget gates can be defined as (1), where σ is a sigmoid activation function, x_t is vector of inputs, h_t is hidden vector of previous layer, W_f and b_f are the weights and bias values of the inputs.

$$f_t = \sigma(W_f \cdot [h_t - 1, x_t] + b_f) \quad (1)$$

The input gates can be expressed as (2) below:

$$i_t = \sigma(W_i \cdot [h_t - 1, x_t] + b_i) \quad (2)$$

The candidate cell state value is calculated using the *tanh* activation function, as shown in (3).

$$L_t = tanh(W_c \cdot [h_t - 1, x_t] + b_c) \quad (3)$$

The cell state is then updated using (4).

$$C_t = f_t * C_t - 1 + i_t * L_t \quad (4)$$

The *output gate* determines which information will be passed to the next hidden state and produced as the output of the model. The output gate is formulated as (5).

$$o_t = \sigma(W_o \cdot [h_t - 1, x_t] + b_o) \quad (5)$$

The final hidden state output is calculated as (6).

$$h_t = o_t * tanh\,(C_t) \quad (6)$$

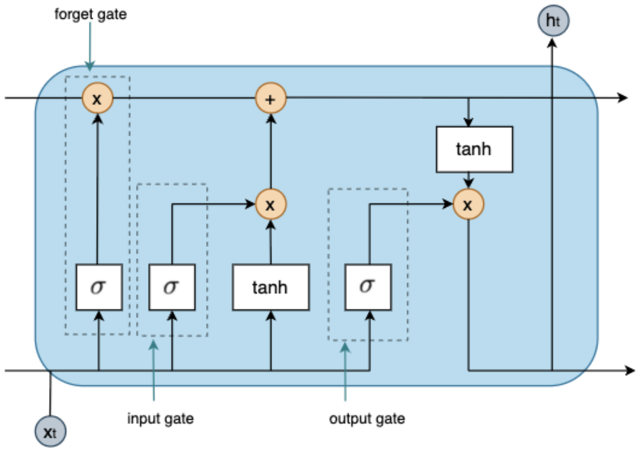

Figure 4. LSTM architecture

2.4.2 GRU

GRU is a development of the RNN architecture [23]. GRU was designed to address the limitations of conventional RNNs, particularly the vanishing gradient problem, while providing a simpler architecture compared to LSTM. Similar to LSTM, GRU is capable of learning temporal dependencies and sequential patterns in time series data. However, GRU uses a simpler structure with fewer parameters, allowing faster training and lower computational complexity while still maintaining competitive forecasting performance [24]. Unlike LSTM, GRU does not use a separate cell state. GRU combines the hidden state and memory mechanism into a unified structure controlled by two main gates, including input gates and update gates. Formulation of GRU can be expressed as following equation:

$$r_t = \sigma(W_{xr}x_t + W_{hr}h_{t-1} + b_r) \quad (7)$$

$$z_t = \sigma(W_{xz}x_t + W_{hz}h_{t-1} + b_z) \quad (8)$$

$$\underline{h}_t = tanh(W_{xh}x_t + W_{hh}(r_t \odot h_{t-1}) + b_h) \quad (9)$$

$$h_t = z_t \odot h_{t-1} + (1 - z_t) \odot \underline{h}_t \quad (10)$$

where r_t, z_t, x_t, h_t respectively defined as reset gate, update gate, input vector and output vector. Similar to LSTM, W represent weight value, b is bias value, σ is a sigmoid activation function, and $tanh$ is the hyperbolic tangent activation function. The architecture of GRU can be seen in Figure 5.

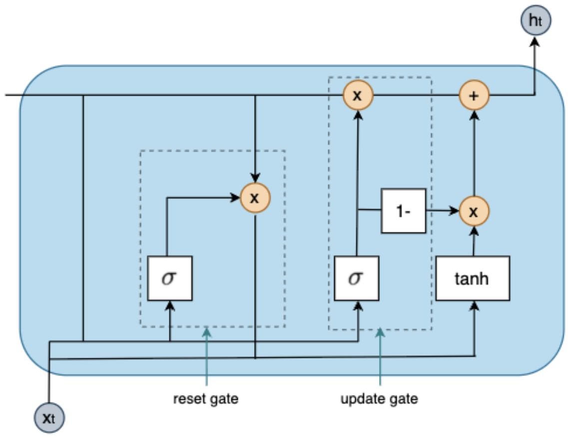

Figure 5. GRU architecture

2.4.3 Preliminary Experiment

Before the main experiments were conducted, a preliminary experiment was performed to identify suitable hyperparameter settings for the LSTM and GRU models using the training and validation datasets. The experiment evaluated several combinations of sequence lengths (7, 14, and 30 days) and hidden unit configurations (64-32 and 32-16) using a two-layer recurrent architecture. Additionally, dropout rates of 0.2 and 0.1 were applied to reduce overfitting and improve model generalization. The configurations were assessed based on forecasting accuracy, training stability, and computational efficiency. The best-performing configuration was subsequently adopted for all experimental scenarios to ensure a fair comparison between LSTM and GRU models and to isolate the impact of temporal feature engineering on forecasting performance.

2.5. Evaluation

The evaluation phase was conducted to measure and compare the forecasting performance of the proposed deep learning models. In this study, the performance of LSTM and GRU models was evaluated under different experimental scenarios, including forecasting with and without temporal feature engineering. The evaluation process was performed using the testing dataset, which was separated chronologically from the training and validation datasets to preserve the temporal characteristics of the time series data. The predicted values generated by each model were compared with the actual commodity prices to assess forecasting accuracy. Several performance metrics used in this study, namely Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE) [25], and also Mean Absolute Percentage Error (MAPE) [26].

To ensure the reliability and stability of the forecasting models, this study performs statistical experiments by repeating each training process multiple times. Deep learning models such as LSTM and GRU involve random initialization of weights and stochastic optimization processes, which may produce slightly different forecasting results in each execution. Therefore, repeated experiments are necessary to evaluate model consistency and robustness. Each experimental scenario was executed five times using the same dataset configuration and hyperparameter settings. The evaluation metrics obtained from each run were then averaged to produce the final performance results. In addition, the standard deviation of each metric was calculated to measure the stability of the forecasting models. The mean value of each metric represents the average forecasting performance, while the standard deviation indicates the consistency of the model across multiple runs. Lower standard deviation values indicate more stable forecasting performance. The statistical experiment used in this study is summarized in Table 3.

Table 3. Statistical Experimental Setup

Component	Configuration
Commodity	Rice, Chili
Forecasting models	LSTM, GRU
Forecasting scenarios	With and without temporal feature engineering
Number of runs	5 runs per scenario
Evaluation metrics	MAE, RMSE, MAPE
Result aggregation	Mean and standard deviation

3. RESULTS AND DISCUSSION

This section presents the evaluation results of the LSTM and GRU models for daily food commodity price forecasting. The evaluation was conducted based on statistical experimental setup. The models were assessed using several performance metrics to measure their forecasting accuracy and predictive capability.

3.1. Preliminary Experiment Result

The preliminary experiment evaluated several combinations of sequence lengths, hidden units, and dropout rates. As a result of the preliminary experiment, the best performing LSTM hyperparameter configuration consisted of two LSTM layers with 64 and 32 hidden units, respectively, a sequence length of 7 days, and dropout rates of 0.2. This configuration achieved the best balance between forecasting accuracy, model stability, and computational efficiency in all subsequent experiments. The GRU configuration consisted of a sequence length of 30, two GRU layers with 64 and 32 hidden units, and a dropout rate of 0.2, achieving the lowest average MAPE (2.44%). Although the configuration achieved the lowest MAPE, the configuration with a sequence length of 7, hidden units of 64–32, and a dropout rate of 0.1 demonstrated substantially lower variability across repeated runs. Therefore, this configuration was selected as the final model due to its superior balance between forecasting accuracy and stability. This preliminary result can be seen in Table 4 and Table 5.

Table 4. Preliminary Experiment Result for LSTM

Sequence Length	**Hidden Unit**	**Dropout**	**MAPE (%)**
7	32-16	0.1	5.64 ± 0.86
		0.2	5.20 ± 0.99
	64-32	**0.1**	**3.79 ± 0.11**
		0.2	4.70 ± 0.79
14	32-16	0.1	7.42 ± 0.63
		0.2	7.11 ± 0.96
	64-32	0.1	4.55 ± 0.08
		0.2	4.87 ± 0.21
30	32-16	0.1	5.69 ± 0.61
		0.2	6.53 ± 0.51
	64-32	0.1	5.03 ± 0.34
		0.2	5.53 ± 0.12

Table 5. Preliminary Experiment Result for GRU

Sequence Length	Hidden Unit	Dropout	MAPE (%)
7	32-16	0.1	3.23 ± 1.43
		0.2	4.33 ± 0.55
	64-32	**0.1**	**2.76 ± 0.12**
		0.2	2.88 ± 0.07
14	32-16	0.1	2.94 ± 1.11
		0.2	3.53 ± 0.47
	64-32	0.1	3.14 ± 0.12
		0.2	3.07 ± 0.15
30	32-16	0.1	4.66 ± 0.91
		0.2	2.59 ± 0.83
	64-32	0.1	3.25 ± 0.12
		0.2	2.44 ± 0.80

3.2. Forecasting Performance Comparison

The first experiment was conducted using historical price data without additional temporal feature engineering. In this scenario, the models relied solely on historical price patterns to forecast future prices. Table 6 shows that GRU consistently outperformed LSTM for both commodities. For rice forecasting, GRU achieved a lower MAPE of 0.49% compared to 1.15% obtained by LSTM. It indicates that the simpler gating mechanism of GRU was sufficient to capture the relatively stable price patterns of rice. Similarly, for chili forecasting, GRU also achieved a lower MAPE of 3.26% than LSTM which recorded a MAPE of 4.67%.

The forecasting results also present a substantial difference between the two commodities. Both models achieved lower forecasting errors for rice than for chili. This finding suggests that rice prices exhibit more stable temporal patterns, making them easier to predict. In contrast, chili prices which have higher volatility and more frequent fluctuations, resulting in larger forecasting errors for both deep learning models. Furthermore, in terms of model stability, LSTM produced relatively consistent results across repeated runs, as indicated by the lower standard deviation values. However, despite its higher variability, GRU still achieved better average forecasting accuracy. This result suggests that GRU provides a more effective representation in food commodity price data while maintaining a lightweight architecture.

Table 6. Forecasting Performance of LSTM and GRU Without Feature Engineering

Model	Commodity	MAE	RMSE	MAPE (%)
LSTM	Rice	173.51 ± 29.68	179.23 ± 28.88	1.15 ± 0.20
	Chili	2713.15 ± 78.79	4042.72 ± 91.17	4.67 ± 0.16
GRU	Rice	73.94 ± 52.69	81.08 ± 47.65	0.49 ± 0.35
	Chili	1881.52 ± 506.89	2935.00 ± 470.84	3.26 ± 0.90

3.3. Impact of Temporal Feature Engineering

The next experiment incorporated temporal feature engineering, including lag features, rolling mean, and price difference features. These additional temporal features were designed to help the models better understand historical patterns, local trends, and price movement dynamics. Based on Table 7, GRU consistently outperformed LSTM across both commodities and experimental settings. The lowest forecasting errors were achieved by GRU with temporal feature engineering, resulting in MAPE values of 0.48% for rice and 2.48% for chili.

Furthermore, the impact of temporal feature engineering can be seen in Table 8. These results show that temporal feature engineering consistently improves the performance across all experimental settings. The most significant improvement was observed in rice commodities with the LSTM model (improved 31.30%). This improvement suggests that the additional temporal features provided explicit contextual information that complemented the sequential learning process of LSTM. Without these features, the model relies only on historical price observations to infer temporal dependencies. By incorporating temporal attributes directly into the input space, the LSTM was able to better distinguish recurring seasonal patterns and calendar-related effects, leading to more accurate forecasts for commodities exhibiting strong temporal variations. The GRU model achieved up to 23.93% improvement for chili price forecasting. This result indicates that temporal feature engineering helped the model identify periodic fluctuations that may not have been fully captured through recurrent memory alone. Explicit temporal information provided additional signals that enhanced the model's ability to represent these complex temporal dynamics. In contrast, the improvement for rice forecasting using GRU was marginal (2.04%). A possible explanation is that the baseline GRU model had already learned most of the relevant temporal dependencies from the historical price sequence. The gating mechanism in GRU is effective at retaining long term information while maintaining a simpler architecture than LSTM. Consequently, the additional temporal features contributed limited new information, resulting in only marginal performance gains. This finding suggests that the benefit of temporal feature engineering depends on both the commodity characteristics and the model's inherent capability to extract temporal patterns from sequential data.

Table 7. Forecasting Performance of LSTM and GRU With Temporal Feature Engineering

Model	Commodity	MAE	RMSE	MAPE (%)
LSTM	Rice	119.98 ± 87.35	129.25 ± 88.05	0.79 ± 0.61
	Chili	1910.62 ± 60.46	2921.37 ± 61.26	3.29 ± 0.11
GRU	Rice	72.47 ± 99.43	81.52 ± 99.20	0.48 ± 0.66
	Chili	1431.02 ± 343.48	2284.76 ± 260.71	2.48 ± 0.61

Table 8. Impact of Temporal Feature Engineering

Model	Commodity	MAPE without FE (%)	MAPE with FE (%)	Reduction in MAPE (%)
LSTM	Rice	1.15	0.79	31.30
	Chili	4.67	3.29	29.55
GRU	Rice	0.49	0.48	2.04
	Chili	3.26	2.48	23.93

In terms of model stability, the results show that the standard deviation of rice commodities is still high. So, although the average accuracy increased, the rice forecasting results still showed quite high variation between runs. LSTM with feature engineering exhibited the most stable forecasting performance, achieving in chili commodity with a MAPE of 3.29 ± 0.11%.

The results of this study demonstrate that commodity volatility characteristics influence the effectiveness of forecasting models and temporal feature engineering. Highly volatile commodities such as chili benefited more from additional temporal features compared to stable commodities such as rice. Furthermore, the findings indicate that although deep learning models such as LSTM and GRU are designed to automatically learn temporal dependencies, temporal feature engineering can still improve forecasting performance, particularly for highly fluctuating data. These findings support the use of hybrid forecasting approaches that combine deep learning capabilities with engineered temporal features.

4. CONCLUSION

This study compared the forecasting performance of LSTM and GRU models for daily food commodity price forecasting in Bandung City, representing low volatility (rice) and high volatility (chili) commodities. The experimental results demonstrate that GRU consistently outperformed LSTM across all forecasting scenarios. Without temporal feature engineering, GRU achieved lower forecasting errors than LSTM for both commodities, indicating its effectiveness in capturing temporal dependencies from historical price data.

The study also investigated the impact of temporal feature engineering on forecasting performance. The results show that incorporating temporal features improved the accuracy of both deep learning models. The improvement was particularly significant for chili prices, where forecasting errors were reduced by up to

29.55% for LSTM and 23.93% for GRU. For rice forecasting, temporal feature engineering provided substantial improvement for LSTM (31.30%), while only marginal improvement was observed for GRU, suggesting that GRU was already able to learn most temporal patterns directly from historical price sequences. Among the two models, the LSTM model shows the largest reduction in MAPE after the application of feature engineering.

Overall, the best forecasting performance was achieved by the GRU model with temporal feature engineering, obtaining a MAPE of 0.48% for rice and 2.48% for chili. These findings indicate that GRU is a suitable deep learning architecture for food commodity price forecasting and that temporal feature engineering remains beneficial even in deep learning-based forecasting frameworks, particularly for commodities with high price volatility.

However, this study has several scope limitations. The analysis was conducted using two food commodities with contrasting volatility characteristics and data from Bandung City, so the conclusions should be interpreted within the context of these commodities and market conditions. The study was intentionally designed to isolate and evaluate the contribution of temporal feature engineering in deep learning based forecasting, providing a controlled comparison between LSTM and GRU architectures. Future research should extend this study by validating the proposed approach across a broader range of commodities and multiple geographical regions. A more comprehensive analysis could also investigate the interaction between temporal feature engineering and other external factors such as weather conditions, supply chain indicators, and economic variables, to further improve forecasting performance and support food price monitoring and inflation management.

REFERENCES

[1] W. Z. Siregar, R. P. Wibowo, and E. Siahaan, “Socioeconomic Dimensions of Food Price Fluctuations and Regional Inflation in Indonesia: Insights from Java and Sumatra,” *Baileo: Jurnal Sosial Humaniora*, vol. 3, no. 2, pp. 332–352, Oct. 2025, doi: 10.30598/baileofisipvol3iss2pp332-352.

[2] O. Akinyemi, H. O. Omisore, and J. A. Kupolusi, “Analysis of Food Price Indices and Inflation Rate in Nigeria,” *International Journal of Science for Global Sustainability*, vol. 11, no. 1, pp. 33–37, Mar. 2025, doi: 10.57233/ijsgs.v11i1.779

[3] Rohimuddin and J. L. Panjawa, “The Impact of Food Commodity Prices on Inflation in Bekasi,” *Marginal Journal Of Management Accounting General Finance And International Economic Issues*, vol. 2, no. 1, pp. 193–206, Sep. 2022, doi: 10.55047/marginal.v2i1.376.

[4] Laura Vantania Safitri, M. Ridwansyah, Rosmeli Rosmeli, Erni Achmad, Nurhayani Nurhayani, and Zainul Bahri, “Analysis Contributor Inflation Group ‘Volatile Food’ from Sector Agriculture in Jambi Province,” *International Journal of Economics and Management Research*, vol. 4, no. 2, pp. 609–619, May 2025, doi: 10.55606/ijemr.v4i2.401.

[5] C. J. Anwar, I. Suhendra, A. Srimulyani, V. M. Zahara, R. A. F. Ginanjar, and S. C. Suci, “Food Price and Inflation Volatilities during Covid-19 Period: Empirical Study of a Region in Indonesia,” *WSEAS TRANSACTIONS ON BUSINESS AND ECONOMICS*, vol. 20, pp. 1839–1848, Aug. 2023, doi: 10.37394/23207.2023.20.161.

[6] H. H. Win and T. T. Zin, “Time Series Forecasting of Commodity Prices Using ARIMA, Holt-Winters and LSTM,” in *2025 6th International Conference on Advanced Information Technologies (ICAIT)*, IEEE, Nov. 2025, pp. 1–6. doi: 10.1109/ICAIT68809.2025.11236779.

[7] D. SJ, P. DV, S. MS, and R. SR, “Time series forecasting of price for oilseed crops by combining ARIMA and ANN,” *International Journal of Statistics and Applied Mathematics*, vol. 8, no. 4, pp. 40–54, Jul. 2023, doi: 10.22271/maths.2023.v8.i4a.1098.

[8] C. Liu, “Long short-term memory (LSTM)-based news classification model,” *PLoS One*, vol. 19, no. 5, p. e0301835, May 2024, doi: 10.1371/journal.pone.0301835.

[9] M. Krichen and A. Mihoub, “Long Short-Term Memory Networks: A Comprehensive Survey,” *AI*, vol. 6, no. 9, p. 215, Sep. 2025, doi: 10.3390/ai6090215.

[10] J. Xu, Z. Wang, X. Li, Z. Li, and Z. Li, “Prediction of Daily Climate Using Long Short-Term Memory (LSTM) Model,” *International Journal of Innovative Science and Research Technology (IJISRT)*, pp. 83–90, Jul. 2024, doi: 10.38124/ijisrt/IJISRT24JUL073.

[11] M. Cho, C. Kim, K. Jung, and H. Jung, “Water Level Prediction Model Applying a Long Short-Term Memory (LSTM)–Gated Recurrent Unit (GRU) Method for Flood Prediction,” *Water (Basel)*, vol. 14, no. 14, p. 2221, Jul. 2022, doi: 10.3390/w14142221.

[12] P. Wanda, “GRUSpam: robust e-mail spam detection using gated recurrent unit (GRU) algorithm,” *International Journal of Information Technology*, vol. 15, no. 8, pp. 4315–4322, Dec. 2023, doi: 10.1007/s41870-023-01516-z.

[13] S. Mohsen, “Recognition of human activity using GRU deep learning algorithm,” *Multimed Tools Appl*, vol. 82, no. 30, pp. 47733–47749, Dec. 2023, doi: 10.1007/s11042-023-15571-y.

[14] P. Joshi, J. Bhardwaj, S. Størdal, M. Kolhe, and E. Haugom, “Analysis of Feature Engineering in Deep Learning Architectures for Dynamic Electricity Price Forecasting,” in *2025 International Conference on Artificial Intelligence, Computer, Data Sciences and Applications (ACDSA)*, IEEE, Aug. 2025, pp. 1–5. doi: 10.1109/ACDSA65407.2025.11165808.

[15] B. M. Porto and F. S. Fogliatto, “Enhanced forecasting of emergency department patient arrivals using feature engineering approach and machine learning,” *BMC Med Inform Decis Mak*, vol. 24, no. 1, p. 377, Dec. 2024, doi: 10.1186/s12911-024-02788-6.

[16] Y. Suhanda, I. Kurniati, and S. Norma, “Penerapan Metode Crisp-DM dengan Algoritma K-Means Clustering Untuk Segmentasi Mahasiswa Berdasarkan Kualitas Akademik,” *J. Teknol. Inform. dan Komput. MH Thamrin*, vol. 6, no. 2, pp. 12–20, 2020.

[17] M. Lewinski, “Statistical Forecasting Methods and Machine Learning Models in Hierarchical Forecasting for Supply Chain Applications,” Nov. 2024, pp. 286–295. doi: 10.21741/9781644903315-33.

[18] Ü. Aycel and Y. Santur, “A New moving average approach to predict the direction of stock movements in algorithmic trading,” *Journal of New Results in Science*, vol. 11, no. 1, pp. 13–25, Apr. 2022, doi: 10.54187/jnrs.979836.

[19] R. J. Hyndman, “Moving averages,” in *International Encyclopedia of Statistical Science*, M. Lovric, Ed. Berlin, Heidelberg, Germany: Springer, 2025, pp. 1559–1562.

[20] L. B. V. de Amorim, G. D. C. Cavalcanti, and R. M. O. Cruz, “The choice of scaling technique matters for classification performance,” *Appl Soft Comput*, vol. 133, p. 109924, Jan. 2023, doi: 10.1016/j.asoc.2022.109924.

[21] G. van Houdt, C. Mosquera, and G. Nápoles, "A review on the long short-term memory model," *Artif Intell Rev*, vol. 53, no. 8, pp. 5929–5955, Dec. 2020, doi: 10.1007/s10462-020-09838-1.

[22] X. Wen and W. Li, "Time Series Prediction Based on LSTM-Attention-LSTM Model," *IEEE Access*, vol. 11, pp. 48322–48331, 2023, doi: 10.1109/ACCESS.2023.3276628.

[23] K. Cho, B. van Merrienboer, D. Bahdanau, and Y. Bengio, "On the Properties of Neural Machine Translation: Encoder–Decoder Approaches," in *Proceedings of SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation*, Stroudsburg, PA, USA: Association for Computational Linguistics, 2014, pp. 103–111. doi: 10.3115/v1/W14-4012.

[24] A. Shewalkar, D. Nyavanandi, and S. A. Ludwig, "Performance Evaluation of Deep Neural Networks Applied to Speech Recognition: RNN, LSTM and GRU," *Journal of Artificial Intelligence and Soft Computing Research*, vol. 9, no. 4, pp. 235–245, Oct. 2019, doi: 10.2478/jaiscr-2019-0006.

[25] T. O. Hodson, "Root-mean-square error (RMSE) or mean absolute error (MAE): when to use them or not," Jul. 19, 2022, *Copernicus GmbH*. doi: 10.5194/gmd-15-5481-2022.

[26] G. van Houdt, C. Mosquera, and G. Nápoles, "A review on the long short-term memory model," *Artif Intell Rev*, vol. 53, no. 8, pp. 5929–5955, Dec. 2020, doi: 10.1007/s10462-020-09838-1.