Scientific Journal of Informatics
Vol. 12, No. 4, November 2025

p-ISSN 2407-7658 http://journal.unnes.ac.id/nju/index.php/sji e-ISSN 2460-0040

Impact of Feature Engineering on XGBoost Model for Forecasting Cayenne Pepper Prices

Jasman Pardede[1]*, Anisa Putri Setyaningrum[2], Marisa Premitasari[3], Muhammad Ilyas Al-Fadhlih[4]
[1,2,3,4]Department of Informatics, Institut Teknologi Nasional Bandung, Indonesia

Abstract.
Purpose: Cayenne pepper represents one of Indonesia's key horticultural commodities, widely utilized in both household culinary practices and the food processing industry. Nevertheless, its market price is subject to considerable volatility, driven by factors such as weather variability, limited supply, production costs, and inefficiencies in distribution systems. This price instability generates uncertainty that adversely impacts farmers, traders, and consumers. Consequently, the development of a reliable price forecasting model is crucial to facilitate price stabilization and enable data-driven decision-making across the supply chain. This study aims to investigate the extent to which feature engineering techniques can enhance the predictive performance of the Extreme Gradient Boosting (XGBoost) algorithm in forecasting cayenne pepper prices. Through the integration of lag features, moving averages, and seasonal indicators, the proposed model is expected to more effectively capture market dynamics and provide a robust analytical tool for relevant stakeholders.
Methods: The forecasting model was constructed using the XGBoost algorithm in combination with various feature engineering methods. The dataset consists of daily price records obtained from Bank Indonesia's PIHPS system and meteorological variables sourced from BMKG, encompassing the period between 2021 and 2024. The engineered features include lag variables identified through Autocorrelation Function (ACF) and Partial Autocorrelation Function (PACF) analyses, Simple Moving Averages (SMA), seasonal indicators, and holiday-related variables designed to capture recurring patterns and event-driven price fluctuations. To enhance predictive performance, hyperparameter tuning was conducted using a grid search optimization approach.
Result: The optimal model demonstrated substantial performance improvements under the following hyperparameter configuration: alpha = 0, gamma = 0.3, lambda = 1, learning_rate = 0.05, max_depth = 3, min_child_weight = 3, n_estimators = 200, and subsample = 0.6. The application of feature engineering markedly enhanced the model's predictive capability, increasing the R^2 value by 99.10% while reducing the MAE, RMSE, and MAPE by 72.63%, 71.31%, and 72.04%, respectively. These outcomes signify a notable reduction in forecasting errors and demonstrate the model's improved accuracy.
Novelty: This study integrates multi-level price data with weather and holiday-related features, employing the ACF and the PACF analyses to determine optimal lag values (techniques commonly utilized in statistical modeling). This integration enhances both the accuracy and interpretability of the XGBoost algorithm, thereby providing a practical and effective tool for agricultural price forecasting and market planning.

Keywords: Cayenne pepper, XGBoost, Feature engineering, Lag features, Forecasting
Received August 2025 / **Revised** November 2025 / **Accepted** November 2025

INTRODUCTION

Chili represents one of the principal commodities within Indonesia's horticultural sub-sector, where maintaining supply and price stability is vital to the national economy. The prices of red and cayenne chili are notably volatile, contributing 0.15% and 0.05% to national inflation in 2019, respectively [1]. Such volatility generates adverse conditions for the agribusiness sector, as unpredictable price movements heighten financial risks for both farmers and traders. For smallholder farmers, these fluctuations pose even greater challenges due to limited access to reliable market information, inadequate storage infrastructure, and restricted flexibility in adjusting sales timing to obtain favorable prices [2]. Consequently, income instability discourages sustained investment and long-term planning in chili cultivation [3].

Cayenne pepper, in particular, is among the most extensively utilized spices in Indonesian households and the food processing industry. However, its production instability contrasts with the steadily increasing per capita consumption driven by population growth [4],[5]. Based on PIHPS data, the average price of cayenne pepper in Bandung City reached Rp 97,500 per kilogram between January and November 2024, highlighting the market's sensitivity to both production and distribution dynamics. Several factors contribute to this volatility, including limited supply, climate variability, rising production costs, and the

* Corresponding author.
Email addresses: jasman@itenas.ac.id (Pardede), anisaputrisetyaningrum@itenas.ac.id (Setyaningrum), marisa@itenas.ac.id (Premitasari), m.ilyasalfadlih@gmail.com (Al-Fadhlih)
DOI: 10.15294/sji.v12i4.32157

intricate structure of the distribution network [6],[7]. These factors are reflected in the dataset employed in this study, which comprises price data from various market levels (producer, collector, and retail) alongside daily weather variables such as temperature, humidity, and rainfall. Disruptions in production can trigger sharp price increases, whereas oversupply and weak demand may lead to price declines, ultimately undermining farmers' economic well-being [8],[9].

Under these circumstances, developing an accurate price forecasting model is crucial to facilitate informed decision-making and promote market stability. Machine learning algorithms (particularly the XGBoost model) have recently gained prominence due to their capability to capture nonlinear patterns and complex interdependencies within data. However, to attain optimal performance in time-series forecasting, these models require robust feature engineering processes to transform raw temporal data into structured inputs suitable for supervised learning [10].

Previous studies [11] have demonstrated that the XGBoost algorithm outperforms traditional time-series models such as ARIMA, LSTM, Prophet, and Gradient Boosting Decision Tree (GBDT), especially when augmented with feature engineering techniques. Zhang et al. (2021) further substantiated these findings, showing that XGBoost consistently achieved superior forecasting accuracy in sales volume prediction when feature engineering was applied. Their experiments revealed that XGBoost attained lower RMSE and MAE values while requiring fewer iterations than GBDT, indicating higher predictive precision and greater computational efficiency in processing time-dependent data influenced by external variables such as weather and temperature.

Compared with conventional time-series models like ARIMA, LSTM, Prophet, and GBDT, XGBoost effectively captures nonlinear dynamics and accommodates multivariate inputs. The results of Zhang et al. (2021) reaffirm this capability, demonstrating that XGBoost delivers enhanced accuracy with fewer training iterations than GBDT, thereby confirming its efficiency. These advantages render XGBoost particularly suitable for this study, as cayenne pepper prices often exhibit irregular, seasonal, and weather-dependent fluctuations that can be more accurately represented through engineered features such as lag variables, moving averages, and temperature indicators.

Li (2023) also demonstrated that XGBoost attains higher predictive accuracy than LSTM in stock price forecasting, thereby reinforcing its reliability across various forecasting domains [12]. Similarly, a study on cayenne pepper prices in Jakarta implemented feature engineering techniques using market data from several traditional markets in DKI Jakarta; however, it excluded external variables such as weather conditions and multi-level price structures [13]. In that study, the procedure for selecting lag features was not explicitly described, although the model achieved a notable R^2 value of 0.92. Another investigation [14] utilized XGBoost with feature engineering to forecast rice prices in Indonesia and reported strong predictive performance, yet the analysis remained limited to price-based features without incorporating additional influencing factors. Several other studies have drawn comparable conclusions, highlighting that XGBoost offers a robust foundation for applying machine learning methods to enhance market forecasting accuracy and to formulate effective pricing strategies [15].

Given the persistent volatility of cayenne pepper prices and the multitude of factors influencing them, this study employs the XGBoost algorithm, recognized for its precision and robustness in modeling complex, nonlinear relationships. To improve model performance and uncover essential price dynamics, various feature engineering techniques are incorporated, including seasonal variables (month and day) to capture recurring temporal patterns, SMA to mitigate short-term fluctuations, lag features to represent historical dependencies, and holiday indicators to account for the effects of special events.

Although previous studies have incorporated external factors such as weather conditions into price forecasting, many have not integrated these variables within a comprehensive feature engineering framework. Conversely, research employing XGBoost with feature engineering has seldom included external influences. Moreover, while lag features are frequently utilized in time-series forecasting, earlier studies often neglected the critical process of determining the optimal lag period (the key factor in capturing temporal dependencies and enhancing model accuracy) [16].

To overcome these limitations, this study employs a comprehensive dataset that integrates multi-level cayenne pepper price data (producer, collector, and market) with external variables such as temperature,

humidity, rainfall, and public holidays. Multiple feature engineering techniques are applied, including seasonal variables, SMA, lag features, and event-based indicators. The novelty of this research lies in the determination of optimal lag periods through ACF and PACF analyses (methods traditionally employed in statistical models such as ARIMA but rarely applied to XGBoost) based forecasting.

The primary objective of this study is to develop an accurate and interpretable model for forecasting cayenne pepper prices in Bandung. The proposed model is expected to offer valuable insights for agricultural planning, risk management, and policy decision-making related to market regulation and supply chain operations.

METHODS

The model's performance was assessed through a systematic workflow beginning with data collection and preprocessing to ensure dataset reliability and consistency. Subsequently, feature engineering techniques were employed to construct informative variables, while lag values were identified using the ACF and the PACF analyses to determine the most significant historical dependencies. The engineered features generated from these processes were then utilized to train the XGBoost model. Finally, model performance was evaluated using multiple metrics to assess predictive accuracy and overall effectiveness. The complete business process framework is illustrated in Figure 1.

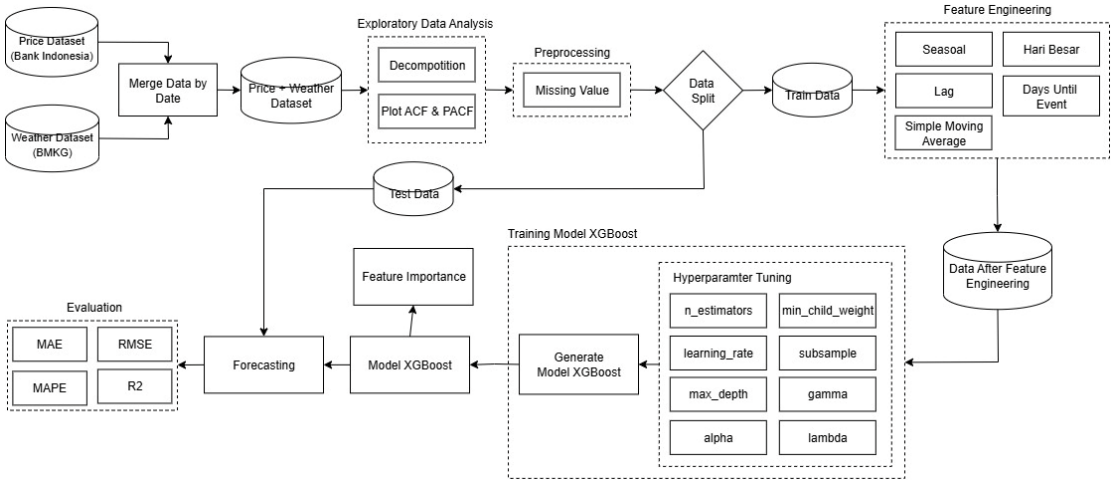

Figure 1. Research Flow

Data Collection

The dataset used in this study is secondary data which includes cayenne pepper price data and external factors. Cayenne pepper price data is obtained from the National Strategic Food Price Information Center (PIHPS) (https://www.bi.go.id/hargapangan), a platform managed by Bank Indonesia, for the period January 2021 to December 2024. External factors such as temperature, humidity, and rainfall are taken from the BMKG Online Data website (https://dataonline.bmkg.go.id/beranda). The data is recorded on a daily basis, resulting in 1,097 observations over the four-year period. After preprocessing and feature engineering, the dataset contains 15 variables including multi-level price data (producer, collector, market) and engineered features such as seasonal indicators, simple moving averages (SMA), lag features, and holiday indicators. Table 1 depicts a data set containing 1,097 entries and 15 variables, along with their complete descriptions presented in Table 2.

Table 1. Dataset

Date	Temperature	Humidity	Rainfall	Producer Prices	Collector Prices	Market Prices
2021-09-01	12400	18000	30000	24.5	71	0
2021-09-01	12400	17000	30000	24.1	69	0
….	….	….	….	….	….	….
2024-09-01	38400	34000	57500	26.1	74	0

Table 2. Description of dataset attributes

No	Attribute	Description	Source
1	Date	Observation date	PIHPS/BMKG
2	Temperature	Daily average temperature (°C)	BMKG
3	Humidity	Daily average relative humidity (%)	BMKG
4	Rainfall	Daily total rainfall (mm)	BMKG
5	Producer Prices	Daily cayenne price at producer level (IDR/kg)	PIHPS
6	Collector Prices	Daily cayenne price at the collector level (IDR/kg)	PIHPS
7	Market Prices	Daily cayenne price at retail market level (IDR/kg)	PIHPS

Autocorrelation

Observations in a time series are often correlated with previous observations, making them non-independent. This relationship is known as autocorrelation or serial correlation. As mentioned, a time series with autocorrelation does not meet the assumptions of standard regression analysis. The ACF is used to check whether the data is stationary and to measure autocorrelation. The ACF displays the correlation between each data point and its previous values at various time intervals (lags), where lag indicates the number of time intervals between two observations. In addition to the ACF, there is also the PACF, which measures the correlation between the current observation and the observation at a specific lag after removing the influence of correlations from shorter lags. For example, the PACF at lag 4 shows the correlation between the value of Y_t and the value at lag 4 (Y_{t-4}) after controlling for the influence of lags 1 through 3 ($Y_{t-1}, Y_{t-2}, Y_{t-3}$) [17].

Preprocessing

Data preprocessing constitutes the preliminary stage of data analysis and machine learning model development. Its primary objective is to clean, transform, and prepare the dataset to ensure it can be effectively utilized by the algorithm. This stage is crucial, as real-world data frequently contains missing values, inconsistencies, and noise that may compromise model accuracy. Common preprocessing activities include managing missing data, removing irrelevant or redundant features, converting categorical variables into numerical representations, and detecting and eliminating outliers [18].

Feature Engineering

Feature engineering refers to the process of modifying or creating new features from raw data to enhance the quality of data representation during model training. The primary goal of this process is to enrich the dataset, thereby enabling the model to identify underlying patterns and relationships more effectively [19]. By incorporating relevant information from historical observations, feature engineering allows models such as XGBoost to capture temporal trends with higher precision [19].

Because most machine learning algorithms cannot be directly applied to time-series data, a common strategy involves transforming the data into a feature vector format that can be processed by conventional predictive models. Constructing this representation is a critical step in time-series learning and exerts a substantial influence on model performance. Although manual feature generation can be complex and time-intensive, effective feature engineering remains essential for enhancing forecasting accuracy [20].

In this study, several feature engineering techniques were employed, including lag features, SMA, seasonal indicators, and holiday-based features. Moreover, the optimal lag period was determined using the ACF and the PACF analyses to ensure that the most relevant historical dependencies were accurately captured.

Extreme Gradient Boosting (XGBoost)

Extreme Gradient Boosting, commonly known as XGBoost, is a machine learning algorithm widely applied to both classification and regression problems. It employs a tree-boosting approach that enables the creation of highly flexible and complex predictive models through specific parameter settings and configurations.

The working principle of XGBoost is similar to that of other Gradient Boosting methods, in which multiple weak learners are combined iteratively to form a stronger and more accurate ensemble model, as shown in Figure 2. The purpose of this combination is to improve performance through gradual training. The prediction results from the previous model, known as residuals or errors, are used as evaluation material to enhance model performance [21].

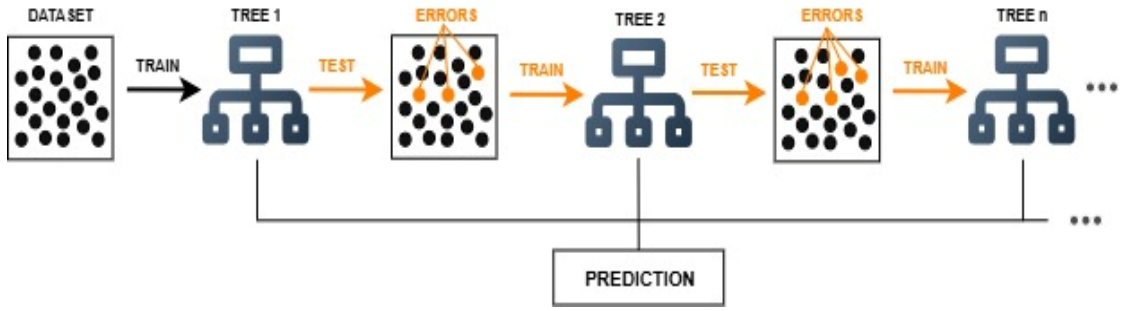

Figure 2. Extreme Gradient Boosting (XGBoost)

The fundamental difference between Extreme Gradient Boosting and the standard Gradient Boosting algorithm is that XGBoost uses a more structured regularization model formalization to control overfitting, which can provide better model performance. In regression, the objective function is used to reduce prediction errors and minimize the loss function and regularization term, as shown in Equation (1):

$$obj(\theta) = \sum_{j=1}^{T_k} L(f_k) + \sum_{j=1}^{T_k} \Omega(f_k) \quad (1)$$

Where $\sum_{j=1}^{Tk} L(f_k)$ is a loss function and $\sum_{j=1}^{Tk} \Omega(f_k)$ is a regulatory term.
The loss function is used to measure the error of a model in predicting data, so that it can be determined whether the model can predict accurately. The formula for the loss function is described in Equation (2). Here, G_{jk} is the first-order gradient statistic of the loss function (sum of gradients), H_{jk} is the second-order gradient statistic of the loss function (sum of Hessians), and W_{jk} is the weight.

$$L(f_k) = \sum_{j=1}^{T_k} \left[G_{jk} W_{jk} + \frac{1}{2} H_{jk} {W_{jk}}^2 \right] \quad (2)$$

The regulation term serves to control how complex the model is. The regularization term applies a penalty mechanism to control model complexity, helping the model prevent overfitting and maintain generalization performance. The details of the regularization formula are presented in Equation (3), where T denotes the total number of leaves in the constructed tree, W represents the weight assigned to each leaf, and γ, λ, and α are constant coefficients that regulate model flexibility.

$$\Omega(f_k) = \gamma T_k + \frac{1}{2} \lambda \sum_{j=1}^{T_k} {W^2}_{jk} + a \sum_{j=1}^{T_k} |W_{jk}| \quad (3)$$

Hyperparameter Tuning
Hyperparameter tuning represents a critical phase in the machine learning pipeline, as it significantly affects a model's performance and generalization capability. In contrast to model parameters, which are automatically learned during training, hyperparameters are predefined values that control the model's overall structure and complexity. The tuning process entails evaluating various combinations of hyperparameters to identify the configuration that delivers optimal performance for a specific task. Proper hyperparameter optimization helps balance bias and variance, thereby mitigating the risks of overfitting and underfitting while enhancing predictive accuracy.

Automated optimization approaches such as grid search, random search, and Bayesian optimization have gained widespread adoption in modern machine learning due to their efficiency in streamlining the tuning process [22]. Effective hyperparameter optimization provides substantial benefits for researchers, data analysts, and industry practitioners by ensuring that models operate at their full potential [23].

In this study, the Grid Search method was employed for hyperparameter optimization. This approach systematically examines all possible combinations of hyperparameter values within predefined ranges, generating a grid of candidate configurations. For each combination, the model is trained and validated to identify the settings that yield the highest predictive performance [24], [25]. Grid Search is particularly advantageous when the search space is relatively limited and precise optimization is required to achieve high model accuracy.

Feature Importance

Identifying the most influential features within a model is fundamental to understanding its predictive mechanisms and interpreting its overall behavior. Feature importance (also referred to as feature detection, feature attribution, or model interpretability) is closely aligned with the statistical concepts of estimation and attribution. This approach assigns a numerical value or metric to each feature, enabling the ranking of features based on their contribution to the model's predictive performance. Such rankings are typically derived by systematically permuting feature values and measuring the corresponding decline in predictive accuracy. Through this process, each feature receives an importance score that facilitates direct comparison [26].

In the XGBoost framework, feature importance can be assessed using three primary metrics: weight, gain, and cover. Weight denotes the frequency with which a feature is used to split data across all trees in the model. Gain represents the average improvement in model accuracy resulting from splits involving that feature, while cover reflects the average number of observations affected by those splits. Collectively, these metrics provide a comprehensive understanding of the relative influence of each feature on the model's predictive outcomes [27].

Time Series Forecasting

Time series forecasting is the process of estimating future values with the highest possible accuracy based on patterns extracted from past observations. This approach not only relies on historical data but also incorporates relevant external information, such as anticipated events or conditions that may influence future patterns [28],[29]. Time series data has several characteristics, such as:

1) Trend: A consistent long-term upward or downward movement observed across data points.
2) Seasonal: Recurring patterns that repeat over fixed intervals, such as monthly or yearly cycles.
3) Cyclical: Fluctuations that occur over longer periods, often associated with transitions between different phases, as seen in business cycles.
4) Random Variate: Irregular variations that cannot be predicted and are often treated as noise.

Evaluation

The model's forecasting performance was evaluated using several metrics, including Root Mean Square Error (RMSE), Mean Absolute Percentage Error (MAPE), Mean Absolute Error (MAE), and the Coefficient of Determination (R^2). These metrics were used to assess the model's accuracy on the test dataset and to provide a comprehensive evaluation of its predictive capability.

$$\text{MAE} = \frac{1}{m}\sum_{i=1}^{m}|X_i - Y_i| \tag{4}$$

The MAE metric is suitable when outliers are considered irrelevant, as it uses the L1 norm, which limits the impact of extreme values and provides a balanced measure of average error. Consequently, MAE provides a stable and bounded performance measure for the model. However, if the test set contains many outliers, the model performance measured by MAE may become suboptimal [30]. The MAE formula is shown in equation (4).

$$\text{RMSE} = \sqrt{\frac{1}{m}\sum_{i=1}^{m}(X_i - Y_i)^2} \tag{5}$$

RMSE measures the square root of the average of the squared differences between predicted and actual values. Unlike MAE, RMSE assigns a greater penalty to large errors (outliers), making it more sensitive to extreme values. A lower RMSE value indicates that the model's predictions are more accurate [30]. The RMSE formula is shown in equation (5).

$$\text{MAPE} = \frac{1}{m}\sum_{i=1}^{m}\left|\frac{Y_i - X_i}{Y_i}\right| x100 \quad (6)$$

MAPE is a regression performance metric with an intuitive interpretation in terms of relative error. Due to its definition, it is recommended for cases where sensitivity to relative changes is more important than sensitivity to absolute change [30]. The MAPE formula is shown in equation (6).

$$R^2 = 1 - \frac{\sum_{i=1}^{m}|X_i - Y_i|^2}{\sum_{i=1}^{m}|\bar{Y} - Y_i|^2} \quad (7)$$

The coefficient of determination represents the proportion of variance in the dependent variable that can be explained or predicted by the independent variables in the model [30]. The R^2 formula is shown in equation (7).

RESULT AND DISCUSSION

This chapter presents the results from each stage of the research methodology, including exploratory data analysis, preprocessing, feature engineering, model training, feature importance, and performance evaluation. Each step is discussed to provide insight into its contribution to the final forecasting model.

An initial data analysis process that aims to understand the characteristics, structure, and important components of the dataset. The decomposition of the daily cayenne market price time series into trend, seasonal, and residual components is presented in Figure 3.

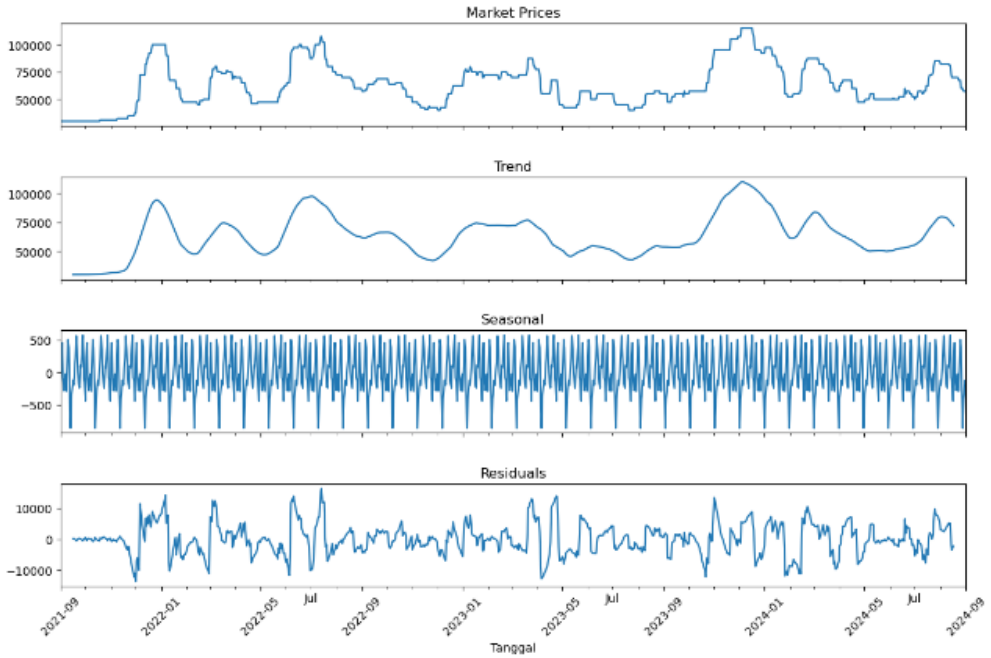

Figure 3. Time series decomposition of market prices into trend, seasonal, and residual components

In Figure 3 shows the time series decomposition of daily cayenne market prices into trend, seasonal, and residual components. It can be seen that the market price trend experienced several phases of increase and decrease in the time span 2021 to 2024. The seasonal component exhibits a consistent and recurring pattern, indicating a distinct seasonal cycle. In contrast, the residual component displays relatively random variations, though without extreme fluctuations. These characteristics suggest that the cayenne pepper market price data contains identifiable patterns that can be further analyzed and modeled. The results of the ACF and PACF analyses of daily cayenne pepper market prices are presented in Figure 4.

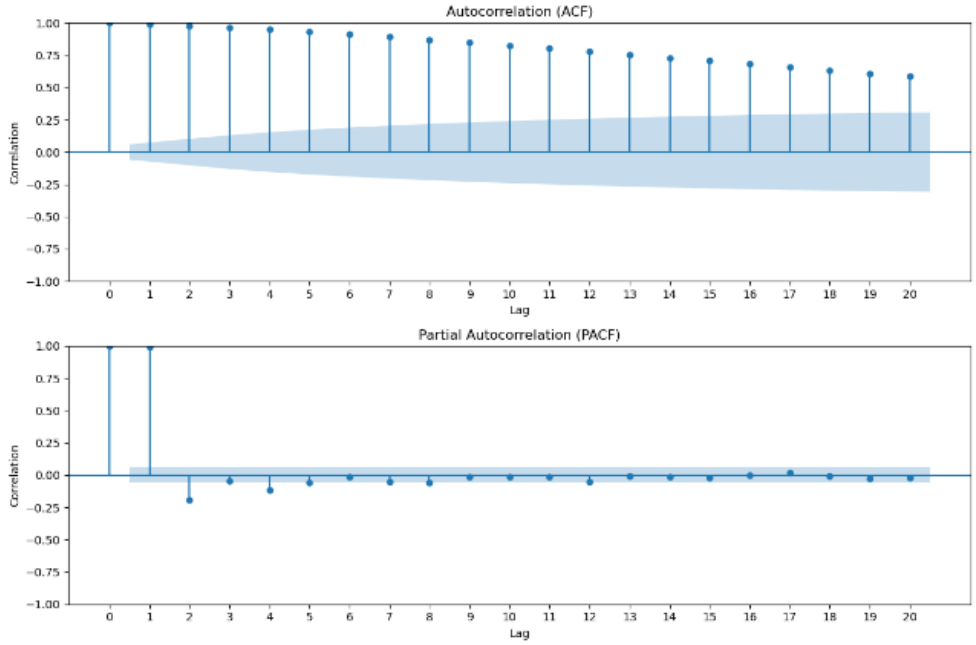

Figure 4. ACF and PACF of market prices

Based on Figure 4, the ACF plot displays several lags that exceed the confidence threshold, indicating the presence of total correlations, including indirect effects. In contrast, the PACF plot highlights the direct relationships between the variable and its lagged values. At a 95% confidence level, only lag 1 and lag 2 are found to be statistically significant. Although lag 4 slightly surpasses the threshold, its contribution to the model is negligible. Consequently, lag 1 and lag 2 were selected as the most relevant features for further modeling.

Prior to model training, a preprocessing stage was carried out to ensure data quality and consistency. Missing values in the price data were addressed using the forward-fill technique to maintain temporal continuity, while missing entries in the weather data were handled through interpolation to preserve natural temporal variations. The detailed preprocessing outcomes are summarized in Table 3.

Table 3. Summary of Missing Value Handling Techniques for Price and Weather Data

Before Missing Value Handling				After Missing Value Handling			
Rainfall	Producer Price	Collector Price	Market Price	Rainfall	Producer Price	Collector Price	Market Price
0	32900	81000	97500	0	32900	81000	97500
NaN	NaN	NaN	NaN	0.3	32900	81000	97500
NaN	NaN	NaN	NaN	0.6	32900	81000	97500
0.9	32900	81000	97500	0.9	32900	81000	97500

Table 3 displays the dataset before and after the treatment of missing values. It is evident that missing entries in the rainfall data were addressed through interpolation, whereby NaN values were replaced with estimated values to maintain temporal consistency. In contrast, missing price data were handled using the forward-fill method to ensure continuity within the time series.

Feature engineering was subsequently performed to enhance the model's predictive capability by incorporating additional relevant variables, as summarized in Table 4. The engineered features include the SMA to smooth short-term fluctuations, lag variables to capture recent price dynamics, and seasonal indicators such as day, month, year, day of the year, and week of the year. In addition, dummy variables

were generated for national and religious holidays, and a "Days Until Event" feature was introduced to account for potential price increases preceding major holidays.

Table 4. Feature Engineering Results

Date	…..	Market Price	Day	Month	Year	Day Of Year	Week Of Year	Lag 1	Lag 2	SMA 2	SMA 3	Holiday	Days Until Event
04-09-2021	…..	35000	4	9	2021	246	35	30000	40000	35000	61666	False	27
05-09-2021	…..	40000	5	9	2021	247	35	35000	30000	32500	35000	False	26
06-09-2021	…..	45000	6	9	2021	248	35	40000	35000	37500	35000	False	25
07-09-2021	…..	40000	7	9	2021	249	35	45000	40000	42500	40000	False	24
08-09-2021	…..	35000	8	9	2021	250	35	40000	45000	42500	41666	False	23

Each record in Table 4 is generated based on date and market price information. For example, on September 4, 2021, the values for the Day, Month, and Year columns are directly derived as 4, 9, and 2021, respectively. The DayOfYear corresponds to the 246th day of the year, while the WeekOfYear represents the 35th week. The Lag 1 and Lag 2 columns contain price data from September 3 (30,000) and September 2 (40,000), respectively. The SMA-2 value reflects the average of the preceding two days (35,000), whereas SMA-3 denotes the average price for September 3, 2, and 1 (61,666). The Public Holiday column is marked False, and the Days Until Event feature holds a value of 27, indicating the number of days remaining until the subsequent holiday. The XGBoost model was subsequently trained through a structured procedure, commencing with hyperparameter optimization via the Grid Search method to determine the optimal parameter configuration, as summarized in Table 5.

Table 5. Hyperparameter Tuning Results for XGBoost Model

Feature Combination	Parameters
Without Feature Engineering	alpha: 0, gamma: 0.0, lambda: 3, learning_rate: 0.05, max_depth: 4, min_child_weight: 4, n_estimators: 100, subsample: 0.5
1 Feature (Lag)	alpha: 2, gamma: 0.0, lambda: 1, learning_rate: 0.05, max_depth: 4, min_child_weight: 1, n_estimators: 200, subsample: 0.7
2 Feature (Lag + SMA)	alpha: 2, gamma: 0.1, lambda: 1, learning_rate: 0.07, max_depth: 4, min_child_weight: 2, n_estimators: 100, subsample: 0.7
3 Feature (Lag + SMA + Holiday)	alpha: 0, gamma: 0.0, lambda: 1, learning_rate: 0.07, max_depth: 3, min_child_weight: 2, n_estimators: 200, subsample: 0.8
4 Feature (All Feature Engineering)	alpha: 0, gamma: 0.3, lambda: 1, learning_rate: 0.05, max_depth: 3, min_child_weight: 3, n_estimators: 200, subsample: 0.6

Grid search was conducted separately for each feature combination, starting from the raw data (no feature engineering) to the inclusion of one to four engineered features. The optimal combination of hyperparameters for each feature group was determined based on cross-validation results, as presented in Table 5. After identifying the best configuration, the model was retrained using the complete training dataset and subsequently evaluated on the test set to assess its overall performance.

Next, we tried combinations of 1 to 4 features, including Lag, SMA, Seasonal, and Holidays in Table 6 to see their effect on model improvement. The features used in these combinations are the result of the

previous feature engineering process, and each combination was paired with optimal hyperparameters identified through grid search as detailed in Table 5.

Table 6. Model Performance with Different Feature Combinations

Feature Combination	MAE	RMSE	MAPE	R^2
Without Feature Engineering	7020	9316	10.59%	0.4808
1 Feature	2154	3007	3.54%	0.9460
2 Feature	1978	2753	3.14%	0.9547
3 Feature	1978	2734	3.16%	0.9553
4 Feature	1921	2672	2.96%	0.9573

All combination results are shown in Table 6. The initial model without engineered features only produced an R^2 value of 0.4808. The addition of one Lag feature improved the model performance, with the R^2 reaching 0.9460 and the MAPE decreasing to 3.54%. The combination of 2 features (Lag and SMA) produced an R^2 of 0.9547, then slightly increased for 3 features (Lag, SMA, and Public Holiday) with an R^2 of 0.9553. The best overall model was obtained with a combination of four features, namely Lag, SMA, Seasonal, and Public Holiday, with the highest R^2 value of 0.9573 and the lowest MAPE of 2.96%. Therefore, the final model utilized a combination of these four feature groups, as this configuration provided the highest accuracy and offered more comprehensive information coverage for capturing seasonal patterns and the influence of public holidays.

After the training process was completed, the model was further analyzed to determine the most influential features in predicting cayenne pepper prices. This analysis, illustrated in Figure 5, highlights the variables that contributed most significantly to model performance and were most frequently used during training.

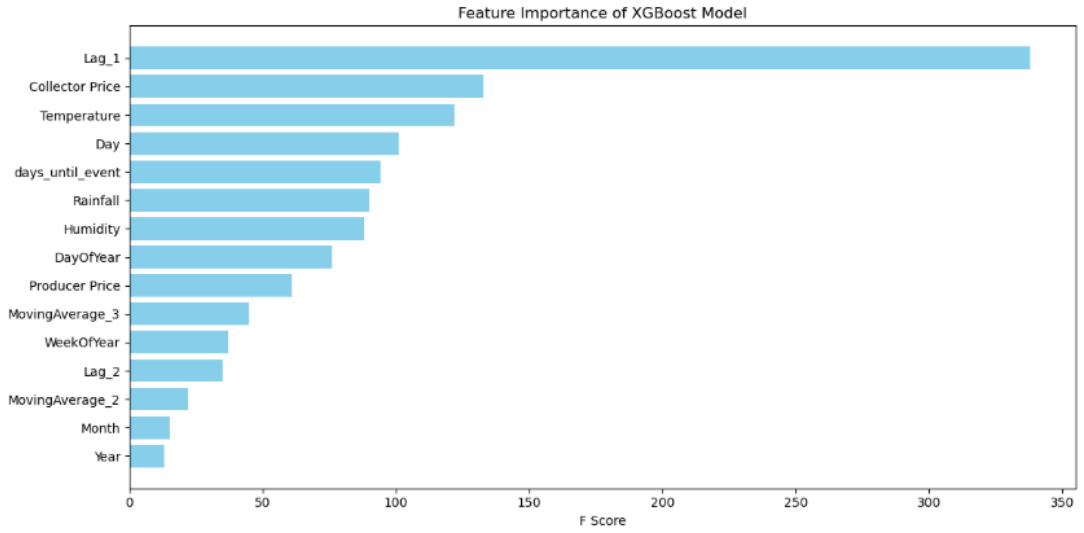

Figure 5. Feature Importance

As illustrated in Figure 5, Lag_1 emerges as the most influential feature, aligning with fundamental principles of time-series forecasting, wherein past observations (particularly the previous day's price) serve as strong predictors of future values. Collector Price ranks second, reflecting the influence of price transmission within the distribution chain on market-level pricing. Temperature follows, underscoring the direct effect of weather conditions on harvest quality and supply stability. Temporal features such as Day and Days Until Event also contribute significantly, indicating short-term seasonal variations and price surges preceding public holidays.

Other weather-related variables, including Rainfall and Humidity, remain relevant due to their impact on cayenne pepper production and transportation logistics. Additional factors such as MovingAverage_3, Lag_2, and Producer Price provide further insights into market dynamics. Meanwhile, time-based features such as WeekOfYear, DayOfYear, Month, and Year exhibit smaller yet meaningful contributions, enabling the model to capture broader seasonal and annual patterns.

To evaluate the impact of feature engineering on model performance, a comparative analysis was conducted between a baseline model without engineered features and an enhanced model incorporating four engineered features (Lag, SMA, Seasonality, and Holiday) as summarized in Table 7.

Table 7. Model Performance Comparison with and without Feature Engineering

Feature Combination	MAE	RMSE	MAPE	R^2
Without Feature Engineering	7020	9316	10.59%	0.4808
With Feature Engineering	1921	2672	2.96%	0.9573

As shown in Table 7, feature engineering markedly improved the model's forecasting accuracy. The R^2 value rose from 0.4808 to 0.9573, representing a 99.10% improvement in explained variance. Meanwhile, MAE, RMSE, and MAPE decreased by 72.63%, 71.31%, and 72.04%, respectively, showing a substantial reduction in forecasting errors and a notable boost in model performance.

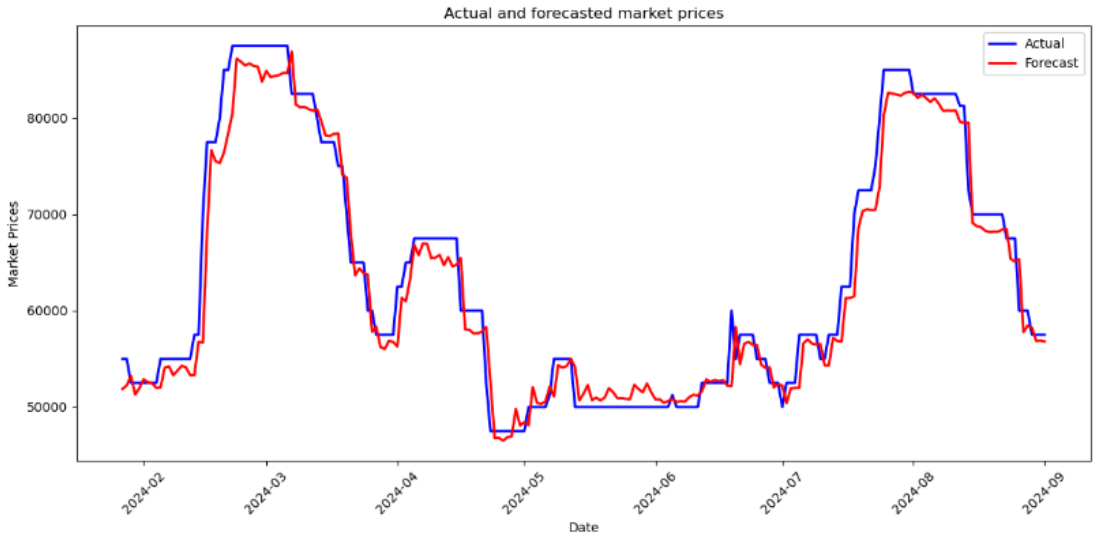

Figure 6. Actual vs. Forecasted Daily Cayenne Market Prices

The visualization presented in Figure 6 illustrates that the forecast line closely tracks actual price movements, particularly during peak periods. The model effectively captures price surges while maintaining stability during normal price intervals. The narrow gap between the predicted and actual lines demonstrates that the model successfully learns both short-term fluctuations and seasonal trends. These findings confirm that the application of feature engineering substantially improves the predictive accuracy of the XGBoost model for daily cayenne pepper prices in Bandung. The integration of lag features, SMA, seasonal indicators, and holiday variables enables the model to more effectively capture short-term variations, recurring seasonal cycles, and the influence of special events.

In comparison with prior studies, these findings further validate the superiority of XGBoost combined with feature engineering over traditional time-series models. For example, [11] demonstrated that XGBoost with engineered features outperformed ARIMA, LSTM, and Prophet in sales forecasting. Similarly, [14] employed XGBoost for rice price prediction in Indonesia but primarily focused on generating forecasts without evaluating performance on test data, thereby limiting the comparability of the results. In contrast,

[13] applied feature engineering for cayenne pepper price forecasting using market data from several traditional markets in DKI Jakarta; however, that study excluded external factors such as weather conditions and multi-level price structures. Additionally, the lag selection procedure in [13] was not clearly defined, indicating a non-systematic approach. Although that study achieved a relatively high R^2 value of 0.92, the inclusion of comprehensive external variables and a systematic lag selection process in this research resulted in even greater accuracy ($R^2 = 0.9573$) and improved generalization capability.

The identification of Lag_1 as the most influential variable reinforces the time-series principle that recent historical values exert a strong influence on future price movements. Furthermore, upstream price levels (collector and producer) and weather-related variables contributed substantially to forecasting accuracy, underscoring the importance of incorporating multi-source data in agricultural price modeling. Overall, the proposed model exhibits both technical robustness and practical applicability within agricultural supply chain management, serving as an effective tool for mitigating price volatility in strategic food commodities. Its combination of high predictive accuracy and contextual relevance positions it as a valuable component for integration into decision-support systems within the horticultural sector.

CONCLUSION

The findings of this study indicate that the application of feature engineering significantly enhances the forecasting performance of the XGBoost model for cayenne pepper prices. Through hyperparameter optimization using the Grid Search method, the optimal parameter configuration was determined to include alpha = 0, gamma = 0.3, lambda = 1, a learning rate of 0.05, maximum depth of 3, minimum child weight of 3, 200 estimators, and a subsample ratio of 0.6. The integration of Lag, SMA, Seasonal, and Public Holiday features effectively captured short-term fluctuations and recurring seasonal dynamics influencing price behavior. Model evaluation revealed a substantial improvement in predictive accuracy and a significant reduction in forecasting errors, as reflected by an increase in the R^2 value of 99.10% (from 0.4808 to 0.9573), alongside decreases in MAE, RMSE, and MAPE by 72.63%, 71.31%, and 72.04%, respectively. Feature importance analysis identified Lag_1 as the most influential variable, followed by Collector Price, Temperature, Day, and Days Until Event. Collectively, these results confirm that the application of feature engineering techniques considerably strengthens predictive performance and provides a practical framework for addressing price volatility in strategic food commodities in Indonesia. Future research could extend this framework by incorporating additional external variables such as transportation costs and market demand indicators, as well as by validating the model across other regions or commodities to assess its generalizability.

REFERENCES

[1] M. Fajar and S. Nonalisa, "Forecasting Chili Prices Using TBATS," *Int. J. Sci. Res. in Multidisciplinary Studies*, vol. 7, no. 2, pp. 1–5, 2021.

[2] H. E. Wibowo, R. R. Novanda, R. Ifebri, and A. Fauzi, "Overview of the Literature on the Impact of Food Price Volatility," *AGRITROPICA : Journal of Agricultural Sciences*, vol. 6, no. 1, pp. 22–32, 2023, doi: 10.31186/j.agritropica.6.1.22-32.

[3] H. E. Wibowo, R. Ifebri, and A. Fauzi, "Comparison of chili price volatility in crucial province of Indonesia," *Journal of Agri Socio Economics and Business*, vol. 05, no. 1, pp. 89–104, 2023, doi: 10.31186/jaseb.05.1.89-104.

[4] Sumaryanto *et al.*, "Determining Drivers of Chili Farming Participation: Insights from Upper Citarum Watershed, Indonesia," *International Journal of Design and Nature and Ecodynamics*, vol. 19, no. 2, pp. 581–590, 2024, doi: 10.18280/ijdne.190224.

[5] M. C. Zamzami and Nucke Widowati Kusumo Projo, "Nowcasting of Chili Pepper (Capsicum frutescens L.) Prices in East Java Province Using Multi-Layer Perceptron Method," *Proceedings of The International Conference on Data Science and Official Statistics*, vol. 2023, no. 1, pp. 2–12, 2023, doi: 10.34123/icdsos.v2023i1.274.

[6] E. E. Jeksen, A. Asnah, and D. Dyanasari, "Prospects for Increasing Production and Supply Chain of Cayenne Pepper in Indonesia," *Agro Bali : Agricultural Journal*, vol. 7, no. 3, pp. 944–956, 2024, doi: 10.37637/ab.v7i3.1965.

[7] M. N. E. Brahmana, Sahara, and N. K. Hidayat, "Price Volatility Analysis of Red and Cayenne Pepper of Java Islands during Covid-19 Pandemic," *Journal of Economics, Finance and Accounting Studies*, vol. 4, no. 4, pp. 11–18, 2022, doi: 10.32996/jefas.2022.4.4.2.
[8] M. H. Zaman, D. Wahyuningsih, R. Yuwono, and Y. Nugroho, "The Response of Farmer Welfares Amidst Food Prices Shock and Inflation in the Province of East Java," *International Research Journal of Economics and Management Studies*, vol. 3, no. 12, pp. 244–251, 2024, doi: 10.56472/25835238/IRJEMS-V3I12P129.
[9] M. R. Matondang, B. Krisnamurthi, and H. Herawati, "Price Fluctuations and Volatility of National Strategic Food Commodities in Indonesia," *Agrisocionomics: Jurnal Sosial Ekonomi Pertanian*, vol. 8, no. 1, pp. 134–146, 2024, doi: 10.14710/agrisocionomics.v8i1.17753.
[10] J. Brownlee, *Introduction to Time Series Forecasting with Python*, V1.9. 2020.
[11] L. Zhang, W. Bian, W. Qu, L. Tuo, and Y. Wang, "Time series forecast of sales volume based on XGBoost," *Journal of Physics: Conference Series*, vol. 1873, no. 1, 2021, doi: 10.1088/1742-6596/1873/1/012067.
[12] E. Mulyati, D. Hamidin, and M. N. Fauzan, "Kelayakan Teknologi Iot Untuk Kebun Hidroponik Holtikultura Menggunakan Hydropo 4.0 Di Perkebunan Alam Pasundan, Jawa Barat," *J@ti Undip: Jurnal Teknik Industri*, vol. 16, no. 2, pp. 109–115, 2021, doi: 10.14710/jati.16.2.109-115.
[13] D. Riando, A. Afiyati, and U. M. Buana, "IMPLEMENTATION OF XGBOOST ALGORITHM TO PREDICT THE SELLING PRICE OF CAYENNE PEPPERS IN," vol. 4, no. 09, pp. 741–749, 2024, doi: 10.59188/eduvest.v4i9.3784.
[14] Iqbal and N. A. Prasetyo, "MODELING AND PREDICTING INDONESIA RICE PRICES USING HYPERPARAMETER," pp. 459–470, 2024, doi: 10.31967/prmandala.v5i0.1226.
[15] K. Cai and M. R. Rodavia, "XGBoost Analysis based on Consumer Behavior," *Frontiers in Computing and Inteligent Systems*, vol. 5, no. 2, pp. 85–89, 2023, doi: 10.54097/fcis.v5i2.12974.
[16] O. Surakhi *et al.*, "Time-lag selection for time-series forecasting using neural network and heuristic algorithm," *Electronics (Switzerland)*, vol. 10, no. 20, pp. 1–22, 2021, doi: 10.3390/electronics10202518.
[17] D. C. Smith *et al.*, "Sudden Shift to Telehealth in COVID-19: A Retrospective Cohort Study of Disparities in Use of Telehealth for Prenatal Care in a Large Midwifery Service," *Journal of Midwifery and Women's Health*, vol. 69, no. 4, pp. 522–530, 2024, doi: 10.1111/jmwh.13601.
[18] Rofik; Jumanto, "Evaluation of Ridge Classifier and Logistic Regression for Customer Churn Prediction on Imbalanced Telecommunication Data," vol. 12, no. 2, pp. 311–326, 2025, doi: 10.15294/sji.v12i2.24620.
[19] P. Duboue, *The Art of Feature Engineering: Essentials for Machine Learning*. 2020. doi: 10.1017/9781108671682.
[20] T. Verdonck, B. Baesens, M. Óskarsdóttir, and S. vanden Broucke, "Special issue on feature engineering editorial," *Machine Learning*, vol. 113, no. 7, pp. 3917–3928, 2024, doi: 10.1007/s10994-021-06042-2.
[21] W. Anggraeni *et al.*, "Prediction of Dengue Fever Outbreak Based on Climate and Demographic Variables Using Extreme Gradient Boosting and Rule-Based Classification," *SeGAH 2021 - 2021 IEEE 9th International Conference on Serious Games and Applications for Health*, 2021, doi: 10.1109/SEGAH52098.2021.9551900.
[22] A. Mehdary, A. Chehri, A. Jakimi, and R. Saadane, "Hyperparameter Optimization with Genetic Algorithms and XGBoost: A Step Forward in Smart Grid Fraud Detection," *Sensors*, vol. 24, no. 4, 2024, doi: 10.3390/s24041230.
[23] A. M. Yasser A. Ali, Emad Mahrous Awwad, Muna Al-Razgan, "Hyperparameter Search for Machine Learning Algorithms for Optimizing the Computational Complexity," *Neural Processing Letters*, vol. 11, no. 2, pp. 10569–10587, 2023, doi: 10.1007/s11063-023-11339-5.
[24] T. N. Tran and T. A. Nguyen, "Hyperparameter Optimization for Deep Learning Modeling in Short-Term Load Forecasting," *International Journal of Electrical and Computer Engineering Systems*, vol. 16, no. 6, pp. 443–450, 2025, doi: 10.32985/ijeces.16.6.2.
[25] M. Y. Shams, A. M. Elshewey, E. S. M. El-kenawy, A. Ibrahim, F. M. Talaat, and Z. Tarek, "Water quality prediction using machine learning models based on grid search method," *Multimedia Tools and Applications*, vol. 83, no. 12, pp. 35307–35334, 2024, doi: 10.1007/s11042-023-16737-4.
[26] A. M. Musolf, E. R. Holzinger, J. D. Malley, and J. E. Bailey-Wilson, "What makes a good prediction? Feature importance and beginning to open the black box of machine learning in genetics," *Human Genetics*, vol. 141, no. 9, pp. 1515–1528, 2022, doi: 10.1007/s00439-021-02402-z.

[27] B. Zhang, Y. Zhang, and X. Jiang, "Feature selection for global tropospheric ozone prediction based on the BO-XGBoost-RFE algorithm," *Scientific Reports*, vol. 12, no. 1, pp. 1–10, 2022, doi: 10.1038/s41598-022-13498-2.
[28] H. Jorge and S. Domingos, *Feature Engineering Automation for Time Series Analysis*, no. January. 2021.
[29] Z. Wang *et al.*, "Is Mamba effective for time series forecasting?," *Neurocomputing*, vol. 619, 2025, doi: 10.1016/j.neucom.2024.129178.
[30] 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 Computer Science*, vol. 7, pp. 1–24, 2021, doi: 10.7717/PEERJ-CS.623.