Journal of Soft Computing Exploration

Homepage: htttps://shmpublisher.com/index.php/joscex

p-ISSN: 2746-7686
e-ISSN: 2746-0991

.

Comparative evaluation of deep learning models for dried corn price prediction in east java

Antika Zahrotul Kamalia[1], Choiriyatun Nisa Latansa[2], Zaenur Rozikin[3], Hemdani Rahendra Herlianto[4], Shiza Hassan[5]

[1,2,3,4]Department of Informatics Engineering, Universitas Pelita Bangsa, Indonesia

[5]Department of Computer Science and Information Technology, NED University of Engineering and Technology, Pakistan

Article Info

Article history:

Received Mar 16, 2026
Revised Mar 26, 2026
Accepted Apr 07, 2026

Keywords:

Deep learning
Dry shelled corn price
East java
Time-series forecasting
Naïve benchmark

ABSTRACT

Forecasting dry shelled corn prices was important for supporting decision-making by farmers, traders, feed industries, and local governments. This study comparatively evaluated several deep learning models, namely Long Short-Term Memory (LSTM), Gated Recurrent Unit (GRU), Convolutional Neural Network 1D (CNN1D), Temporal Convolutional Network (TCN), and Transformer, for predicting dry shelled corn prices in East Java. Classical benchmark models, namely naïve, drift, and simple exponential smoothing (SES), were also incorporated into the experimental design. Using daily price data from 2020 to 2024, a 30-day lookback window, and multivariate features derived from price movements, calendar variables, and rolling statistics, model performance was assessed using MAE, RMSE, MAPE, sMAPE, and R^2. The results showed that the naïve baseline achieved the best overall performance on the 2024 test set, while TCN was the strongest among the evaluated deep learning models. TCN obtained RMSE of 176.95 and $R^2$of 0.6895, whereas the naïve baseline achieved RMSE of 20.06 and $R^2$of 0.9960. Overall, all deep learning models were outperformed by the naïve persistence benchmark, indicating that greater model complexity did not automatically improve forecasting accuracy on this highly persistent price series.

Corresponding Author:

Antika Zahrotul Kamalia,
Department of Informatics Engineering,
Universitas Pelita Bangsa, Indonesia,
Jl. Inspeksi Kalimalang No.9, Cibatu, Cikarang Sel., Kabupaten Bekasi, Jawa Barat 17530, Indonesia.
Email: antika.kamalia@pelitabangsa.ac.id
https://doi.org/10.52465/joscex.v7i2.48

1. INTRODUCTION

Corn is a strategic commodity in the national food system because it serves not only as a food source but also as a key input for the feed industry [1], [2]. Statistics Indonesia reported that in 2023, the national harvested corn area reached 2.48 million hectares, with 19.99 million tons of dry-shelled corn produced at 28% moisture content, and East Java was among the provinces with the largest harvested corn area in Indonesia. In 2024, the national harvested area was estimated to increase to 2.58 million hectares, with 15.21 million tons of dry shelled corn production at 14% moisture content [3]. Despite this production capacity, domestic corn prices remained

under considerable pressure. A policy brief from the Ministry of Agriculture noted that producer-level corn prices had been above the reference price since March 2023, while in East Java, the corn price in February 2024 increased by 22.93% compared with December 2023 [4]. This condition encouraged the government to set the Government Purchasing Price (HPP) for corn at the farmer level at IDR 5,500/kg in 2025 to stabilize prices while protecting farmers' welfare [5]. These conditions highlight the practical importance of forecasting dry shelled corn prices, particularly in East Java, as one of Indonesia's major corn-producing regions.

Previous studies have shown that deep learning approaches are increasingly used for agricultural commodity price forecasting because they can capture nonlinear relationships and temporal dependencies that are difficult to model using classical approaches [6], [7]. However, recent benchmarking literature emphasizes that the superiority of deep learning models is not universal. In particular, rigorous forecast evaluation requires comparison against simple and classical benchmarks, especially the naïve or persistence forecast, which can be highly competitive for volatile or near-random-walk series. Studies also show that while complex models may provide gains on more predictable data, their advantage can disappear on low-predictability economic and financial time series. Therefore, rigorous benchmarking is essential to determine whether improvements in forecasting accuracy are genuine and practically meaningful [8]–[10]. Gu and Li showed that an LSTM model with external market-related variables improved corn futures price forecasting accuracy, indicating that LSTM is promising for modeling corn-related price series [11]. A recent systematic review on food price prediction also confirmed that LSTM-based and hybrid deep learning models often outperform traditional models, although the results remain highly dependent on the forecasting horizon, commodity type, input variables, and preprocessing strategy [12]–[14]. In Indonesia, Triyadi showed that an LSTM model could forecast dried shallot prices in Wonosobo with reasonably good accuracy, suggesting that deep learning has potential for modeling agricultural and food commodity price series in Indonesia [15]. In addition, a study in East Java by Nensi et al. showed that Bi-LSTM outperformed LSTM and CNN-LSTM across four food commodities, particularly on more volatile commodities such as red chili and shallots [16].

Although these findings demonstrate the promise of deep learning, the available empirical evidence also indicates that no single architecture is universally superior across all time-series settings. LSTM and GRU are widely recognized for handling sequential dependencies, TCN provides long effective memory through dilated convolutions, and Transformer-based models rely on self-attention to capture long-range dependencies [17], [18]. However, model superiority remains data-dependent, meaning that different commodity characteristics may lead to different forecasting performance. Therefore, model selection should be based on direct comparative evaluation using the target price series rather than architectural popularity alone.

Based on the reviewed studies, prior research has mostly focused on national food prices, monthly agricultural commodity prices, corn futures, or other commodities in East Java, so direct evidence comparing several deep learning model families on daily dry shelled corn price data in East Java remains limited. This study, therefore, aims to comparatively evaluate LSTM, GRU, CNN1D, TCN, and Transformer for one-step-ahead forecasting of dry shelled corn prices in East Java using daily data from 2020 to 2024. The main contribution of this study is to provide a local forecasting benchmark that compares deep learning models with classical baseline models, including the naïve benchmark, under a consistent preprocessing pipeline, chronological evaluation, and statistical testing of forecast accuracy differences.

2. METHOD

Research Design

The research workflow illustrates the main stages used to build and compare forecasting models for dry shelled corn price prediction in East Java, starting from data preparation and ending with comparative model evaluation and statistical testing. The workflow consists of Data Collection, Preprocessing and Feature Engineering, Sequence Construction and Data Split, Model Training and Prediction, Baseline Forecasting Models, and Performance Evaluation and Statistical Comparison.

Figure 1. Research flow

As shown in Figure 1, all forecasting models were developed and evaluated under the same chronological pipeline. This design ensures that performance differences mainly reflect model behavior rather than

differences in data treatment, and it allows a fair comparison between deep learning architectures and simple benchmark models.

Dataset and Variables

The dataset consists of daily dry-shelled corn price data from 2020 to 2024 in East Java, obtained from SISKAPERBAPO (Information System on Availability and Price Development of Staple Foods in East Java), organized in a multi-sheet yearly Excel file. The core variables used in the implementation are DATE, PREVIOUS PRICE, CURRENT PRICE, PRICE CHANGE (IDR), and PRICE CHANGE (%). Overall, the dataset contains 1,827 daily observations. Although this size is adequate for one-step-ahead forecasting experiments, it remains relatively limited by deep learning standards, which provides important context for the model configurations adopted in this study and for the interpretation of the comparative results [9], [10]. This variable structure is consistent with the East Java commodity monitoring system, which explicitly lists dry shelled corn as one of the monitored commodities [19]. In this study, the prediction target is defined as the CURRENT PRICE at time t, while the price-change variables are treated as explanatory inputs.

Under this formulation, the problem is modeled as a multivariate time-series regression task in which the model receives a historical feature window and outputs the estimated next target value. In notation,

$$y\hat{}t = f_\theta(X_t) \quad (1)$$

where $X_t$denotes the historical feature window and θdenotes the model parameters. This setup is consistent with supervised sequence forecasting practice.

Data Preprocessing

The preprocessing stage begins with date parsing, chronological sorting, and numeric cleaning so that price and percentage-change fields are stored in a consistent format. Missing values generated at the beginning of the engineered-feature stage are handled using backfilling. This choice was made because the missing values are systematically generated by lagged and rolling transformations at the start of the series, when the required historical window is not yet available. In this setting, forward-filling is not appropriate for the earliest observations because no valid values precede them, whereas interpolation would impose synthetic, smoothed values that may alter the local temporal structure of the price series. Backfilling was therefore preferred as a simple, consistent method for completing the initial observations while preserving the implemented forecasting pipeline, as is commonly recommended in time-series preprocessing practices [9]. This step is required to ensure that all observations can later be converted into fixed-length input sequences. This step was required to ensure that all observations could later be converted into fixed-length input sequences. After basic cleaning, the data were standardized using z-score normalization [20]. For a feature x, the standardized value was computed as:

$$z = \frac{x-u}{s} \quad (2)$$

where u is the mean and $s$is the standard deviation. In this implementation, the scaling parameters are estimated only from the training subset and then applied to the validation and test subsets. This follows the scikit-learn recommendation to avoid fitting preprocessing steps on unseen evaluation data, thereby reducing the risk of data leakage.

Feature Construction

In line with the research flow, the Preprocessing and Feature Engineering stage produces a shared multivariate feature set for all models. Based on the notebook implementation, the features are price, chg_rp, chg_pct, dow_sin, dow_cos, month_sin, month_cos, roll_mean_7, roll_std_7, roll_mean_30, roll_std_30, log_ret, and abs_chg. These features were designed to represent price level, short-term movement, calendar effects, and local trend and volatility. Since chg_rp, chg_pct, log_ret, and abs_chg are all derived from the same underlying daily price change, some degree of redundancy is acknowledged, although each transformation captures a slightly different aspect of price dynamics. The 30-day lookback window was selected empirically to balance capturing recent temporal dependence with preserving a sufficient number of training samples.

Calendar variables are encoded cyclically using sine and cosine so that periodic patterns such as weekday and month are not treated as ordinary linear scales. The cyclical encodings for weekday d_t and month m_t are defined as follows:

$$\text{month\_sin}_t = \sin\left(\frac{2\pi m_t}{12}\right), \text{month\_cos}_t = \cos\left(\frac{2\pi m_t}{12}\right). \quad (3)$$

Rolling statistics are computed as

$$\text{roll\_mean}_{k,t} = \frac{1}{k}\sum_{i=0}^{k-1} p_{t-i}, \quad (4)$$

$$\text{roll\_std}_{k,t} = \sqrt{\frac{1}{k-1}\sum_{i=0}^{k-1}(p_{t-i} - \bar{p}_t)^2}, \quad (5)$$

with $k \in \{7,30\}$. Log return and absolute change are computed as

$$\text{log\_ret}_t = \ln(p_t) - \ln(p_{t-1}), \text{abs\_chg}_t = p_t - p_{t-1}. \quad (6)$$

These transformations directly match the implemented feature-engineering pipeline.

The Sequence Construction and Data Split stage then transforms the multivariate table into supervised sequences using a 30-day lookback window. For each time point t, $X_t = [x_{t-30}, x_{t-29}, \ldots, x_{t-1}], y_t = p_t$. Thus, each model uses the previous 30 observations to forecast the target-day price. This sliding-window representation is consistent with common approaches in sequence learning and time-series forecasting[21].

Deep Learning Models

This study compares five deep learning architectures: LSTM, GRU, causal CNN1D, TCN, and Transformer encoder. LSTM is included because it was specifically introduced to mitigate long-term dependency problems in recurrent neural networks. GRU is used as a more compact gated recurrent alternative that still models sequential dependencies effectively. Both are widely relevant to forecasting because they process time dependence directly through recurrent structures [22], [23].

For convolution-based modeling, the study employs causal CNN1D and TCN. A causal CNN1D captures local temporal patterns without using future information, whereas TCN extends this design through dilated causal convolutions and residual connections, enabling a longer effective memory in sequence modeling. Recent forecasting studies and reviews show that such convolution-based temporal architectures remain competitive for time-series tasks because they preserve temporal order while efficiently modeling both short- and long-range dependencies [23], [24].

The final architecture is a Transformer encoder, included because self-attention can model relationships across time positions without explicit recurrence [25]. To preserve temporal order, the inputs are projected to a latent representation and enriched with sinusoidal positional encoding, following the original Transformer design.

Baseline Forecasting Models

In addition to the deep learning models, this study formally incorporates three classical benchmark models, namely naïve, drift, and simple exponential smoothing (SES). The naïve model, also known as the persistence benchmark, predicts the next value using the most recent observed value, and is formulated as

$$\hat{y}_{t+1|} = y_t \quad (7)$$

where $\hat{y}_{t+1|t}$ denotes the forecast for time $t+1$ made at the time t, and y_t is the most recent observed price. The drift model extends the naïve benchmark by extrapolating the historical trend between the first and most recent observations in the training history. The SES model produces forecasts through exponentially weighted averaging of past observations, assigning greater weight to more recent prices. These baseline models are included because forecasting literature consistently recommends that any complex forecasting method should be evaluated against simple benchmarks before practical conclusions are drawn.

Experimental Setup

Following the research notebook, all deep learning models are implemented in Python with PyTorch, while the benchmark models are implemented using standard time-series forecasting procedures. All models are evaluated on the same chronological split: observations up to 30 September 2023 for training, from 1 October 2023 to 31 December 2023 for validation, and from 1 January 2024 onward for testing. This design ensures that both deep learning models and classical benchmarks are compared fairly under the same out-of-sample forecasting setting.

The notebook configuration sets lookback = 30, maximum epochs = 30, batch size = 64, learning rate = 10^(-3), early-stopping patience = 7, and seed = 42. At the architecture level, LSTM and GRU use one recurrent

layer with 64 hidden units and 0.1 dropout; CNN1D uses 64 channels and kernel size 3; TCN uses 4 dilation levels with 64 channels and kernel size 3; and the Transformer uses d_model = 64, 4 heads, 2 encoder layers, feed-forward dimension 128, and 0.1 dropout. Final predictions are inverse-transformed to the original price scale before evaluation.

No systematic hyperparameter search was conducted; instead, a fixed configuration was used to maintain comparability across models under the same experimental setting. Therefore, some observed performance differences may partly reflect suboptimal settings for certain architectures rather than architectural differences alone. In addition, the validation period covers only approximately three months (about 92 observations), so the early-stopping mechanism may be somewhat sensitive to short-term fluctuations in validation loss.

Evaluation Metrics

The Performance Evaluation stage uses MAE, RMSE, MAPE, sMAPE, and R^2so that model performance is assessed from multiple perspectives. The MAE and RMSE express error on the original price scale, MAPE and sMAPE provide relative error views, and R^2measures the proportion of explained variance. Scikit-learn documents MAE, MAPE, and RMSE as non-negative losses with an ideal value of 0, while R^2has an ideal value of 1 and can be negative for poor models [26]. Forecasting literature also notes that percentage-based measures, especially MAPE and sMAPE, should be interpreted carefully under certain data conditions.

Comparative Analysis Scheme

The final stage combines Performance Evaluation and Comparative Analysis. To ensure fairness, all architectures are tested using the same dataset, feature set, lookback window, train–validation–test split, and scaling procedure. Therefore, any performance difference is interpreted mainly as a consequence of architectural differences rather than different data treatments. This is consistent with good benchmarking practice, which emphasizes a stable and leakage-controlled evaluation pipeline [9], [27].

In this implementation, the main ranking criterion is test-set RMSE, supported by MAE, MAPE, sMAPE, and R^2as complementary indicators. RMSE is prioritized because it penalizes larger forecast errors more heavily, while the other metrics provide supporting views of absolute error, relative error, and overall goodness of fit [26].

Statistical Comparison of Forecast Accuracy

To assess whether the observed differences in forecasting accuracy are statistically meaningful, this study complements point estimates of error metrics with statistical comparison procedures. Forecast accuracy differences between selected model pairs are evaluated using the Diebold–Mariano (DM) test on the 2024 test set. The DM test is applied to forecast errors in order to examine whether one model significantly outperforms another in terms of predictive accuracy. In addition, bootstrap confidence intervals are reported for the main error metrics to provide uncertainty estimates around the observed performance values. Statistical comparison is particularly important in this study because several models show relatively close error values, while others differ substantially from both the best deep learning model and the classical benchmark models.

3. RESULTS AND DISCUSSIONS

Descriptive Analysis of Corn Price Data

The exploratory results show that the East Java dry shelled corn price dataset covers the period from 1 January 2020 to 31 December 2024, with 1,827 daily observations and no missing calendar dates. After cleaning, the daily price series has a mean of IDR 7,183.78/kg, a standard deviation of 594.76, a median of IDR 7,196/kg, a minimum of IDR 4,271/kg, and a maximum of IDR 8,859/kg. Visually, the daily series is not flat: the 2020–2021 period is relatively lower and more stable, followed by a gradual increase during 2022–2023, with the highest levels appearing in early 2024 before a gradual correction toward the end of the year.

Overall, the series exhibits characteristics of persistence, level shifts, and short-term shocks. These properties are important for forecasting because they suggest that recent observations carry strong predictive information, while sudden changes may be difficult to anticipate. As a result, simple forecasting approaches, particularly naïve or persistence-based models, may perform competitively against more complex models in this setting.

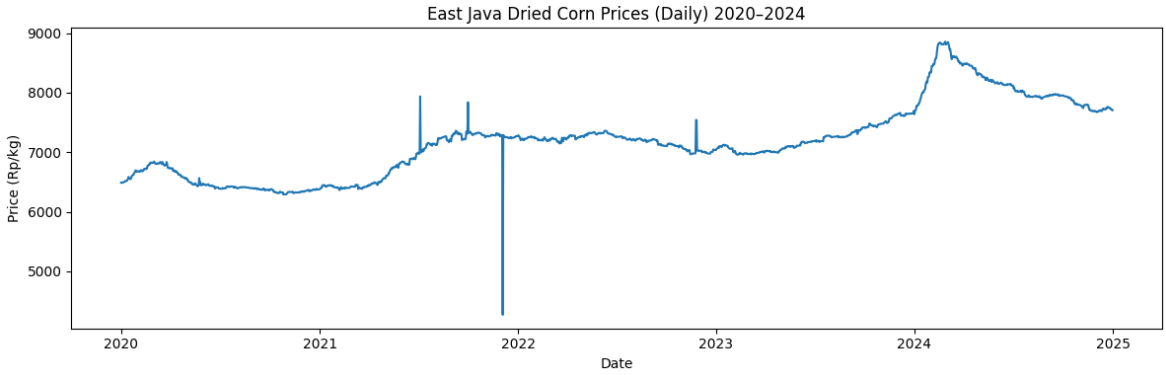

Figure 2. Daily dry shelled corn prices, 2020–2024

As shown in Figure 2, the price histogram indicates that most observations are concentrated around IDR 6,500–7,500/kg, although the distribution still extends toward higher-price regions. The monthly mean plot further highlights level shifts across years, while the daily log-return plot reveals several sharp positive and negative spikes. Taken together, these results suggest that the series contains a combination of persistence, level shifts, and short-term shocks, making it a non-trivial forecasting target.

Preprocessing Results

The preprocessing stage produced a more consistent modeling dataset. Numeric cleaning successfully corrected price values that could initially be read as "6.491" into 6491, and the main columns CURRENT PRICE, PREVIOUS PRICE, PRICE CHANGE (IDR), and PERCENTAGE CHANGE showed no missing values after cleaning. Feature engineering then generated 13 shared input features, including price-related variables, cyclical calendar variables, 7-day and 30-day rolling statistics, log returns, and absolute change. After that, the dataset was transformed into 1,797 supervised sequences with a 30-day lookback window. For reproducibility, the chronological split was defined explicitly as follows: the training set includes sequences up to 30 September 2023, the validation set covers 1 October 2023 to 31 December 2023, and the test set begins on 1 January 2024.

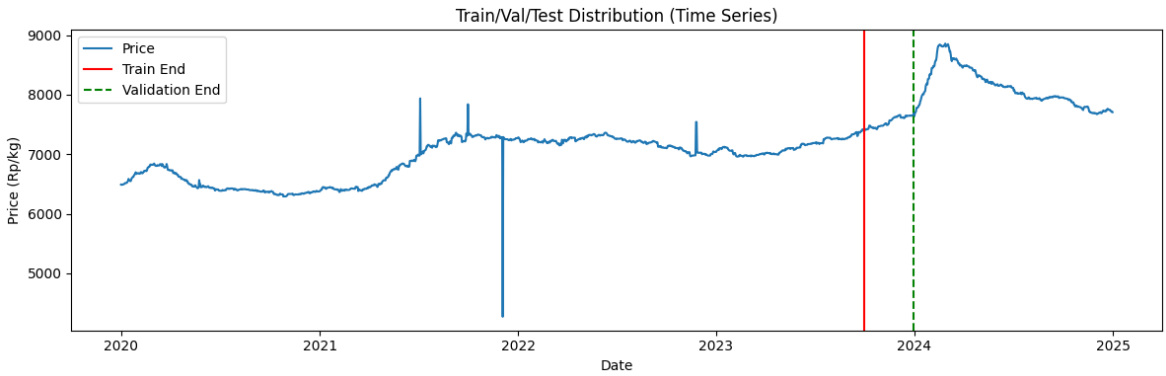

Figure 3. Train–validation–test distribution

As shown in Figure 3, The chronological split produced 1,339 training sequences, 92 validation sequences, and 366 testing sequences. Such a split is important in forecasting because accuracy should be assessed on genuinely future observations rather than on data seen during fitting; evaluation on new data provides a more valid picture of forecasting performance than in-sample errors alone.

Comparative Performance of Deep Learning and Baseline Models

This section presents the predictive performance of all evaluated deep learning and baseline models on the test set in order to identify the most suitable forecasting approach for dry shelled corn prices in East Java.

The evaluation uses MAE, RMSE, MAPE, sMAPE, and R^2, so that the comparison reflects not only absolute and relative forecast errors but also the explanatory performance of each model.

Tabel 1. Predictive performance of deep learning and baseline models on the test set

Model	MAE	RMSE	MAPE	sMAPE	R2	Group
Naive	12.333333	20.055796	0.150572	0.150619	0.996012	Baseline
Drift	12.559642	20.074744	0.153368	0.153396	0.996004	Baseline
SES	31.98724	50.559661	0.389346	0.390265	0.974653	Baseline
TCN	132.11433	176.94919	1.605681	1.628325	0.689535	DeepLearning
CNN1D	121.51763	216.32237	1.457599	1.487829	0.536	DeepLearning
GRU	414.52515	478.56127	5.009878	5.177317	-1.27086	DeepLearning
LSTM	517.81616	592.13401	6.261752	6.521252	-2.4766	DeepLearning
Transformer	566.22345	635.65409	6.858919	7.160027	-3.00642	DeepLearning

As shown in Table 1, the naïve benchmark achieved the best overall performance, with the lowest MAE, RMSE, MAPE, and sMAPE, as well as the highest R^2This indicates that the dry shelled corn price series in East Java is highly persistent, so that a simple one-step persistence forecast is extremely difficult to outperform. Among the evaluated deep learning architectures, TCN achieved the best performance in terms of RMSE and R^2, while CNN1D remained relatively competitive and even produced lower MAE and MAPE than TCN. In contrast, GRU, LSTM, and Transformer yielded substantially larger forecast errors and negative R^2values, indicating weaker predictive performance on the test data. Overall, these results show that although TCN was the strongest deep learning model, none of the deep learning architectures in this study outperformed the benchmark models, particularly the naïve persistence model[26], [28].

Statistical Comparison of Forecast Accuracy

To complement the point estimates in Table 1, forecast accuracy differences were further examined using the Diebold–Mariano (DM) test for selected model pairs. This analysis was conducted to determine whether the observed differences in forecasting performance were statistically meaningful, particularly for the comparisons between the best deep learning model and its closest competitors or benchmark models.

Tabel 2. The diebold–mariano result

Model A	Model B	DM stat_MSE	p_value MSE	mean_loss_diff MSE(A-B)	Better_model MSE
TCN	CNN1D	-5.311516025	1.89E-07	-15484.35254	TCN
TCN	Naive	10.00202104	0	30908.78125	Naive
TCN	Drift	9.998172052	0	30908.02086	Drift
TCN	SES	9.94075277	0	28754.73687	SES
CNN1D	Naive	7.795563072	6.77E-14	46393.13377	Naive
LSTM	Naive	17.65056522	0	350220.4446	Naive
GRU	Naive	16.4271128	0	228618.6596	Naive
Transformer	Naive	18.73013002	0	403653.8773	Naive
Model_A	**Model_B**	**DM_stat_MAE**	**p_value_ MAE**	**mean_loss_diff_ MAE(A-B)**	**Better_model MAE**
TCN	CNN1D	2.527866547	0.01189698	10.59668636	CNN1D
TCN	Naive	20.13382967	0	119.7809925	Naive
TCN	Drift	20.03155977	0	119.554684	Drift
TCN	SES	19.57459232	0	100.1270863	SES
CNN1D	Naive	12.01283922	0	109.1843062	Naive
LSTM	Naive	34.06192706	0	505.4828301	Naive
GRU	Naive	32.65115328	0	402.1918078	Naive
Transformer	Naive	37.11271321	0	553.8900807	Naive

As shown in Table 2, the comparison between TCN and CNN1D indicates different conclusions depending on the loss function used. Under the MSE-based DM test, TCN performs better than CNN1D, whereas under the MAE-based comparison, CNN1D shows a lower mean absolute loss. This result is consistent with Table 1, where TCN achieved better RMSE and R^2, while CNN1D remained competitive in some error measures.

More importantly, the comparisons between TCN and the benchmark models show that the naïve, drift, and SES baselines all outperform TCN, as indicated by the positive mean loss differences and the better-model assignments in Table 2. Among these, the naïve benchmark shows the strongest advantage over TCN, confirming that the best deep learning model in this study still does not surpass the simplest persistence-based forecast on the 2024 test set. Similar results are also observed when CNN1D, LSTM, GRU, and Transformer are compared with the naïve model, where the benchmark model is consistently identified as superior. Overall, these statistical results reinforce the main finding of this study that simple benchmark models, especially the naïve model, provide significantly stronger predictive performance than the evaluated deep learning architectures for this highly persistent price series.

Tabel 3. Bootstrap CI result

Model	RMSE_CI95	MAE_CI95
LSTM	[559.9612, 626.2450]	[489.1937, 547.8723]
GRU	[450.2840, 507.5150]	[391.0352, 440.0176]
CNN1D	[187.3672, 243.7050]	[103.6618, 140.9551]
TCN	[158.5347, 194.2696]	[119.6003, 144.8795]
Transformer	[603.2800, 670.5384]	[537.6763, 596.9736]
Naive	[17.2402, 22.9208]	[10.7377, 14.1531]
Drift	[17.2909, 22.9382]	[10.9870, 14.3513]
SES	[44.5067, 56.4531]	[28.3468, 36.0908]

As shown in Table 3, the bootstrap confidence intervals also reinforce the main finding of this study. The naïve, drift, and SES benchmarks consistently show much lower error ranges than the evaluated deep learning models, indicating that their superiority is robust on the 2024 test set. While TCN remained the strongest deep learning architecture, its confidence intervals were still far above those of the benchmark models, confirming that no deep learning model outperformed the simple persistence-based baseline in this highly persistent price series.

Actual vs Predicted Price Visualization

The actual-versus-predicted visualization focuses on the comparison between the best deep learning model and the strongest benchmark model on the 2024 test set. In this study, TCN represents the best-performing deep learning architecture, while the naïve model represents the best overall benchmark. The visualization is intended to provide a clearer picture of how both approaches follow the actual movement of dry shelled corn prices over time.

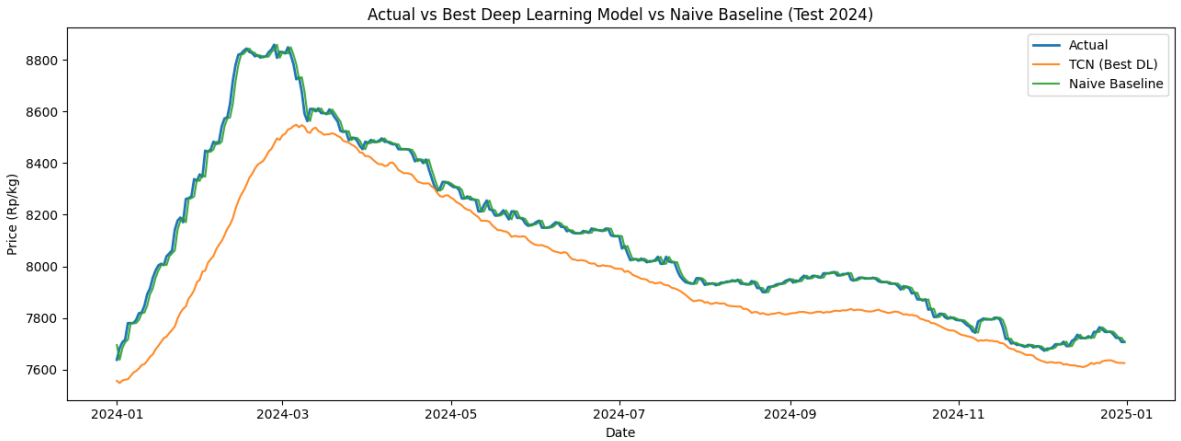

Figure 4. Actual vs. predicted prices: TCN and naïve baseline

As shown in Figure 4, both models were able to follow the broad direction of price movement during 2024, particularly the downward trend observed after the early-year peak. However, the naïve benchmark

tracked the actual price path much more closely than TCN throughout most of the test period, which is consistent with the error metrics reported in Table 1. The figure also shows that TCN tended to underestimate prices during the sharp increase in early 2024, whereas the naïve model remained much closer to the actual series. Overall, this visualization supports the quantitative findings that TCN is the best among the evaluated deep learning models, but not superior to the simple persistence benchmark.

Discussion of Findings

The most important finding of this study is that the naïve benchmark outperformed all evaluated deep learning models on the 2024 test set. This result indicates that the East Java dry shelled corn price series is highly persistent, so that using the most recent observed value already provides an exceptionally strong one-step-ahead forecast. Therefore, the main contribution of this study is not only the comparison among deep learning architectures, but also the demonstration that increased model complexity does not automatically lead to improved forecasting accuracy for highly persistent commodity price series.

Within the deep learning group, TCN achieved the strongest performance, particularly in terms of RMSE and R^2. This is plausible because TCN combines causal convolution with dilated convolution, allowing it to capture local temporal patterns while maintaining a longer effective memory[29]. The relatively strong performance of CNN1D also suggests that convolution-based architectures are better aligned with the local and persistent structure of this dataset than recurrent and attention-based models.

Another important result is the presence of strongly negative R^2values for GRU, LSTM, and Transformer. These values indicate that the three models performed worse than simply predicting the test-period mean. This severe failure may be explained by several factors, including the high persistence of the price series, the relatively limited sample size for more complex architectures, and the tendency of these models to produce overly smoothed forecasts that fail to track local fluctuations. Recent studies also note that recurrent models may still struggle with long-term dependency learning despite gated designs, while transformer-based models trained on limited target datasets may be constrained by the available information and often require larger-scale pretraining to realize their potential[30], [31]. In other words, these models were not only less accurate than TCN but also unable to capture the dominant short-term persistence that characterizes the series.

Taken together, the findings suggest that forecasting performance in this setting is driven less by architectural sophistication and more by the interaction between model structure and the persistence of the underlying series. Thus, while TCN can be regarded as the strongest deep learning architecture in this study, the naïve benchmark remains the most informative and operationally relevant forecasting reference[31], [32].

Table 4. Comparison with previous studies

Study	Data & Scope	Model	Main Result	Comparison with This Study
Triyadi (2022)	Shallot prices (Indonesia)	LSTM	Good forecasting accuracy	Different: dataset characteristics differ
Gu & Li (2023)	Corn futures (global)	LSTM	Improved accuracy with external variables	Consistent: DL works in a complex/global setting
Nensi et al. (2023)	Food commodities (East Java)	Bi-LSTM	Outperformed CNN-LSTM	Partially consistent: DL works for volatile commodities
This study	Corn price (East Java, daily)	TCN, CNN1D, etc.	Naïve baseline outperformed all DL models	New finding: DL fails under high persistence

Table 4 demonstrates the superiority of deep learning models, particularly LSTM-based architectures, in prior research on agricultural commodity price forecasting. These earlier studies consistently reported strong predictive performance in various commodity markets, though they often relied on data with lower persistence, longer training samples, or additional external variables.

In sharp contrast, this study establishes that, for highly persistent daily corn price data in East Java, a simple naïve benchmark decisively outperforms all evaluated deep learning models. This finding underscores that model performance is fundamentally data-dependent, and results from previous studies cannot be universally applied to different forecasting contexts. This study thereby makes a definitive contribution by empirically demonstrating that deep learning models can fail to surpass simple benchmarks when time series data exhibit strong persistence.

Practical Implications

From a practical perspective, the results suggest that operational corn-price forecasting systems should not automatically adopt the most complex architecture. In this dataset, TCN may be considered the strongest deep learning candidate, particularly when the goal is to explore nonlinear sequence patterns within a flexible forecasting framework. However, for day-to-day short-term forecasting, a simple benchmark such as the naïve model should always be included as a mandatory reference before deploying more complex methods.

For local governments and food agencies, this finding implies that routine daily monitoring may not require a complex deep learning system if a persistence-based benchmark already provides very strong short-term accuracy. For traders and feed-industry actors, more complex models may be worth pursuing mainly in operational settings where the forecasting horizon is extended, external variables such as harvest conditions, logistics, or policy shocks are incorporated, or decision-making depends on detecting turning points rather than simply tracking day-to-day persistence. In practice, a more complex model should be adopted only if it produces a stable and meaningful out-of-sample improvement over the naïve benchmark that is large enough to justify the added costs of implementation, maintenance, and monitoring [33].

Limitations of the Study

This study has several limitations. First, the amount of data is relatively limited by deep learning standards, with 1,827 daily observations and 1,797 supervised samples after sequence construction, so more flexible architectures may not have been fully utilized. Second, the features remain centered on price history and its transformations, while external drivers such as harvest seasonality, rainfall, logistics, feed prices, food inflation, or market policy were not included. Third, although classical benchmark models were formally incorporated in the revised experimental design, the study still focuses on a limited set of baseline methods and does not yet include broader statistical forecasting families such as ARIMA or ETS in a full benchmark comparison.

Another limitation is that statistical testing was conducted only on selected forecast comparisons rather than exhaustively across all possible model pairs. Future research should therefore extend both the benchmark set and the statistical comparison framework, and examine whether richer covariates and larger datasets can enable complex models to outperform simple persistence-based forecasting approaches.

4. CONCLUSION

This study compared LSTM, GRU, CNN1D, TCN, and Transformer for forecasting dry shelled corn prices in East Java using daily data from 2020 to 2024, while also incorporating naïve, drift, and simple exponential smoothing as benchmark models. The results showed that the naïve baseline achieved the best overall forecasting performance on the 2024 test set, whereas TCN was the strongest among the evaluated deep learning models. Overall, no deep learning model outperformed the simplest benchmark method on this highly persistent price series. Thus, the study objective stated in the Introduction was achieved, and the results demonstrated that greater model complexity did not translate into practical forecasting gains over a naïve baseline. From an application perspective, persistence-based forecasting may serve as an efficient operational reference for routine short-term price monitoring. Future research should examine whether richer external variables, multi-step forecasting horizons, and hybrid combinations of deep learning and classical models can provide predictive value beyond simple persistence-based forecasting.

REFERENCES

[1] E. P. Maulidiah, B. Budiantono, A. History, and C. Satisfaction, "Peran Pemerintah Dalam Meningkatkan Volume Ekspor Jagung," *J. Econ.*, vol. 2, no. 1, pp. 2137–2146, 2023.

[2] R. Putri, T. Sjah, and K. Budastra, "Pengembangan Model Agribisns Berkelanjutan Pada Tanaman Jagung sebagai Komoditas Unggulan Pertanian," *J. Econ.*, vol. 4, no. 11, pp. 409–421, 2025.

[3] Badan Pusat Statistik (BPS), "Luas Panen dan Produksi Jagung di Indonesia 2024," *Jakarta, BPS*, 2025. .

[4] M. T. Munthe, M. Asaad, R. Karo, and K. Sitepu, "Dampak Kebijakan Harga Acuan Pembelian Pemerintah dan Harga Input terhadap Produksi Jagung di Indonesia," *J. Sos. Ekon. Pertan.*, vol. 21, no. 2, pp. 39–51, 2023.

[5] Badan Pangan Nasional (BPN), "HPP Jagung di Tingkat Petani Rp 5.500 per Kg Resmi Berlaku," *BPN, Jakarta*, 2025. .

[6] H. A. Anamsyah, I. G. S. M. Diyasa, and A. N. Sihananto, "Perbandingan Model Xgboost, Lstm, Dan Neural Prophet Untuk Prediksi Harga Cabai Rawit Merah Di Jawa Timur," vol. 5, no. 2, pp. 31–39, 2025.

[7] A. Keintjem, B. D. Setiawan, and R. S. Perdana, "Analisis komparatif model arima, lstm, dan gru untuk peramalan harga komoditas pangan di kota malang," vol. 10, no. 1, 2026.

[8] S. Makridakis, E. Spiliotis, and V. Assimakopoulos, "M5 accuracy competition: Results, findings, and conclusions," *Int. J. Forecast.*, vol. 38, no. 4, pp. 1346–1364, 2022.

[9] H. Hewamalage, K. Ackermann, and C. Bergmeir, "Forecast evaluation for data scientists: common pitfalls and best practices," *Data Min. Knowl. Discov.*, vol. 37, no. 2, pp. 788–832, 2023.

[10] N. Beck, J. Dovern, and S. Vogl, "Mind the naive forecast! a rigorous evaluation of forecasting models for time series with low predictability," *Appl. Intell.*, vol. 55, no. 6, 2025.

[11] S. Gu and M. Li, "Forecasting Corn Futures Prices Using the LSTM Model," *Financ. Econ. Res.*, vol. 1, no. 1, pp. 1–16, 2024.

[12] M. Rayhan, A. H. Sidhi, and B. R. Japutra, "Model Prediktif Berbasis Data Twitter Dan Google Trends Untuk Estimasi Harga Komoditas Daging Ayam Dan Telur Di Indonesia," vol. 1, no. 1, pp. 35–46, 2025.

[13] K. Choudhary, G. K. Jha, R. Jaiswal, and R. R. Kumar, "A genetic algorithm optimized hybrid model for agricultural price

.

forecasting based on VMD and LSTM network," *Sci. Rep.*, vol. 15, no. 1, pp. 1–20, 2025.
[14] A. Theofilou, S. A. Nastis, A. Michailidis, T. Bournaris, and K. Mattas, "Predicting Prices of Staple Crops Using Machine Learning: A Systematic Review of Studies on Wheat, Corn, and Rice," *Sustain.*, vol. 17, no. 12, pp. 1–34, 2025.
[15] Agus Triyadi, Adi Suwondo, Dian Asmarajati, Nur Hasanah, and Muhamad Fuat Asnawi, "Prediksi Harga Bawang Merah Kering Di Wonosobo Menggunakan Metode Long Short Term Memory," *STORAGE J. Ilm. Tek. dan Ilmu Komput.*, vol. 3, no. 2, pp. 133–138, 2024.
[16] A. I. E. Nensi, W. Pangesti, N. Syukri, M. Al Maida, and K. A. Notodiputro, "Implementing LSTM-Based Deep Learning for Forecasting Food Commodity Prices with High Volatility: A Case Study in East Java Province," *Proc. Int. Conf. Data Sci. Off. Stat.*, vol. 2025, no. 1, pp. 1032–1041, 2025.
[17] X. Ao, Y. Gong, and A. He, "A Review of Time Series Prediction Models Based on Deep Learning," *IEEE Access*, vol. 13, no. September, pp. 153696–153712, 2025.
[18] M. K. Saravana, M. S. Roopa, J. S. Arunalatha, and K. R. Venugopal, "Transformers for Multivariate Time Series Forecasting: Comprehensive Analysis, Challenges, Research Opportunities and Future Prospects," *IEEE Access*, vol. 14, no. December 2025, pp. 11424–11457, 2026.
[19] Siskaperbapo, "Harga Rata-Rata Provinsi Jawa Timur di Tingkat Konsumen," 2025. .
[20] F. Sohil, M. U. Sohali, and J. Shabbir, "An introduction to statistical learning with applications in R," *Stat. Theory Relat. Fields*, vol. 6, no. 1, pp. 87–87, 2022.
[21] L. Su, X. Zuo, R. Li, X. Wang, H. Zhao, and B. Huang, "A systematic review for transformer-based long-term series forecasting," *Artif. Intell. Rev.*, vol. 58, no. 3, 2025.
[22] D. Preet and S. Sachar, "Time Series Forecasting Using Deep Learning: A Comparative Study of LSTM, GRU, and Transformer Models," *J. Comput. Sci. Technol. Stud.*, vol. 5, no. 1, pp. 74–89, 2023.
[23] H. Wang, X. Zhao, Q. Guo, and X. Wu, "A novel hybrid model by integrating TCN with TVFEMD and permutation entropy for monthly non-stationary runoff prediction," *Sci. Rep.*, vol. 14, no. 1, pp. 1–22, 2024.
[24] R. Elmousaid, N. Drioui, R. Elgouri, H. Agueny, and Y. Adnani, "Accurate short-term GHI forecasting using a novel temporal convolutional network model," *e-Prime - Adv. Electr. Eng. Electron. Energy*, vol. 9, no. July, p. 100667, 2024.
[25] A. Katrompas, T. Ntakouris, and V. Metsis, "Recurrence and Self-attention vs the Transformer for Time-Series Classification: A Comparative Study," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 13263 LNAI, pp. 99–109, 2022.
[26] P. Piotrowski, I. Rutyna, D. Baczyński, and M. Kopyt, "Evaluation Metrics for Wind Power Forecasts: A Comprehensive Review and Statistical Analysis of Errors," *Energies*, vol. 15, no. 24, 2022.
[27] M. M. Sesay, "Leakage-aware benchmarking of temporal data preparation for hourly energy forecasting," *Energy Convers. Manag. X*, vol. 30, no. March, p. 101791, 2026.
[28] Y. Yang, "TCN-QV: an attention-based deep learning method for long sequence time-series forecasting of gold prices," *PLoS One*, vol. 20, no. 5 May, pp. 1–24, 2025.
[29] J. Bai, W. Zhu, S. Liu, C. Ye, P. Zheng, and X. Wang, "A Temporal Convolutional Network–Bidirectional Long Short-Term Memory (TCN-BiLSTM) Prediction Model for Temporal Faults in Industrial Equipment," *Appl. Sci.*, vol. 15, no. 4, pp. 1–20, 2025.
[30] S. M. Al-Selwi, M. F. Hassan, S. J. Abdulkadir, and A. Muneer, "LSTM Inefficiency in Long-Term Dependencies Regression Problems," *J. Adv. Res. Appl. Sci. Eng. Technol.*, vol. 30, no. 3, pp. 16–31, 2023.
[31] S. Chakraborty, I. Delibasoglu, and F. Heintz, "Scaling transformers for time series forecasting: do pretrained large models outperform small-scale alternatives?," *Artif. Intell. Rev.*, vol. 59, no. 2, 2026.
[32] F. Alexandrino *et al.*, "A Novel Method for Time Series Prediction with Small Data: Integrating Data Augmentation, Normalization Techniques, and Machine Learning Fernando," 2026.
[33] R. J. Hyndman, "Forecasting:Principles & Practice," *Proc. Forecast. Work. Univ. West. Aust.*, no. September, pp. 1–138, 2024.