Zero : Jurnal Sains, Matematika, dan Terapan
E-ISSN : 2580-5754; P-ISSN : 2580-569X
Volume 10, Number 1, 2026
DOI: 10.30829/zero.v10i1.27631
Page: 69 - 79

Comparative Study of Hybrid ARIMA-LSTM and CNN-LSTM for Palm Oil Price Forecasting

[1] Rizki Alifah Putri
Study Program of Statistics and Data Science, IPB University, Bogor, Indonesia

[2] Khairil Anwar Notodiputro
Study Program of Statistics and Data Science, IPB University, Bogor, Indonesia

[3] Budi Susetyo
Study Program of Statistics and Data Science, IPB University, Bogor, Indonesia

Article Info

Article history:

Accepted 25 January 2026

Keywords:

ARIMA;
CNN-LSTM;
Crude Palm Oil;
Deep Learning;
Hybrid Models;
Time Series Forecasting.

ABSTRACT

The forecasting of highly volatile time series data remains a significant challenge due to complex, non-linear patterns. This study compared the performance of two hybrid frameworks, ARIMA-LSTM and CNN-LSTM, which were designed to integrate the statistical strengths of traditional models with the computational power of deep learning. In these architectures, the ARIMA component was utilized to extract linear trends, while the LSTM and CNN layers were employed to identify and manage non-linear dynamics within the data. Utilizing 384 monthly palm oil price data points (1993-2024) sourced from FRED, the models were evaluated using the Mean Absolute Percentage Error (MAPE) metric. The results demonstrated that the hybrid CNN-LSTM outperformed the ARIMA-LSTM and individual models, achieving a superior MAPE of 6.69%. These findings indicated that the integration of Convolutional Neural Networks (CNN) with Long Short-Term Memory (LSTM) networks was more effective in capturing the complexities of price fluctuations. Practically, the study concluded that accurate forecasting served as a critical tool for market stabilization, thereby supporting broader goals of financial certainty and ecological sustainability.

Corresponding Author:

Rizki Alifah Putri,
Statistics and Data Science Program Study,
IPB University
Email: rizkialifah@apps.ipb.ac.id

1. INTRODUCTION

Forecasting is the process using past data to make predictions about the future. Most forecasting issues pertain to time series data, which are collected periodically over time. Forecasting horizons are usually divided into three groups: short-term, medium-term, and long-term [1]. Many various methods have been applied to make forecasting, from traditional statistical techniques to modern approaches such as machine learning and deep learning, which provide new ways for analyzing and predicting future trends.

Box and Jenkins came up with the ARIMA (Autoregressive Integrated Moving Average) model in 1970, and it has been the standard for predicting time series ever since. ARIMA and SARIMA (Seasonal Autoregressive Integrated Moving Average) models are often used to work with non-stationary data because they can capture both the trend and seasonal parts of a time series [2]. However, ARIMA has limitations and it cannot handle non-linear relationships and assumes that the error variance is always the same, which isn't always true for real-world data conditions. So, ARIMA often combined with other models to fix these problems.

Forecasting methods have changed a lot in the last few decades, especially since artificial intelligence approaches like machine learning and deep learning become popular. Recurrent Neural Network (RNNs) were introduced as possible option; however, their effectiveness was limited by a challenge known as the vanishing gradient problem [3]. To mitigate this issue, Hochreiter and Schmidhuber in 1997 proposed the Long Short-Term Memory (LSTM) architecture, which improves upon traditional RNNs by enabling the retention of long-term information through its improved memory mechanism [4]. Since then, LSTM has become a popular deep learning model for forecasting time series, showing better results in understanding complex patterns over time [5].

In addition to RNN-based models, Convolutional Neural Networks (CNNs) have also gained prominence in forecasting tasks. These networks are effective because they can handle multi-dimensional data and identify key features, which helps in minimizing the impact of random noise in time series predictions [6].

Forecasting challenges are particularly evident in highly volatile commodities like Crude Palm Oil (CPO). As a strategic global commodity, CPO price fluctuations significantly impact food, energy, and industrial sectors. Inaccurate predictions can lead to severe financial repercussions and weakened risk management for producers. Furthermore, market unpredictability is often associated with unplanned plantation expansion, which potentially contributes to ecological degradation, such as reduced water retention and increased flooding [7]. Therefore, enhancing CPO price prediction accuracy is essential for market stability and may provide a technical basis for more sustainable land-use policies.

Previous literature has explored various hybrid frameworks. Zhang [8] demonstrated that combining linear and non-linear models yielded higher accuracy than using ARIMA or ANN in isolation. However, traditional ANNs lack the memory capacity of modern deep learning to capture long-term dependencies. Subsequent studies, such as the CNN-LSTM model for stock prices [9], utilized CNNs for feature extraction and LSTM for prediction, achieving superior accuracy over single models. Similarly, research on tourism demand [2] showed that hybrid architectures outperform traditional models in short-term tasks but struggle with multi-step non-linear dynamics. Comparative studies [10] also suggest that while LSTM excels with large datasets, ARIMA may remain superior for smaller, univariate data.

Despite these advancements, a significant literature gap persists: most studies stop at validating hybrid models against single structure benchmarks, leaving a lack of direct comparative evidence between different hybrid architectures themselves. There is little clarity on whether a statistically-driven hybrid like ARIMA-LSTM or a purely deep learning hybrid like CNN-LSTM is more effective for the extreme volatility of palm oil prices. Therefore, this study aims to evaluate and compare the performance of these two hybrid models using real data on palm oil prices. By shifting the focus from "hybrid versus single" to a "hybrid versus hybrid" analysis, this research provides deeper methodological insights and superior prediction accuracy for analysis and assist in making informed long-term decisions and shaping relevant policies.

2. RESEARCH METHODS

2.1 Data Preparation

Figure 1 shows the monthly real-world data on global palm oil prices from 1993 to 2024, which includes 384 data points collected from the Federal Reserve Economic Data (FRED) database under the "PPOILUSDM" series, which tracks prices in US dollars per metric ton. The statistics of the data show an average price of 613.1051, with the lowest price being 185.0650 and the highest being 652.7163. As seen in Figure 1, palm oil prices have generally been going up, but the changes in price have become bigger as the prices themselves have increased.

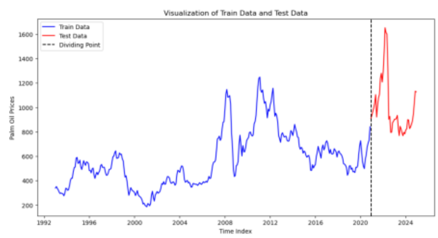


Figure 1. Palm Oil Price Data Plot

Before building our models, we took several steps to prepare the data for analysis:

a. Preprocessing data: a preliminary screening of the "PPOILUSDM" series for this step showed there were no missing values, which means the data flows smoothly over time for the hybrid models. During the preparation of the data, no unusual values were taken out or changed. All changed in prices were kept because they show real market changes, which are important for teaching the models how to deal with actual price changes in the real world.

b. Data splitting: to ensure the reliability of the evaluation reflects the model's adaptability to recent volatility, the split between training and testing sets was determined using Change Point Detection (CPD). We employed the Binary Segmentation algorithm with an L2 cost function to identify significant structural breaks in the time series. The algorithm detected a primary change point at observation index 335. Consequently, the first 335 observations (approximately 87%) were allocated for training, while the remaining 49 observations (approximately 13%), which exhibit the most recent data distribution, were reserved for testing.
c. Stationarity: The Augmented Dickey-Fuller (ADF) test was applied to the training dataset to evaluate the stationarity of the series. This was important because the ARIMA model needs data that has a steady mean and variance. To achieve stationarity in variance, a Box-Cox transformation was implemented, while stationarity in the mean was addressed through the differencing process [11]. These steps helped find the right value for the integrated part (d) of the ARIMA model, which is necessary for reliable time series analysis.
d. Normalization: all price data were normalized using Min-Max scaling to a range of [0,1] to facilitate stable model training and prevent feature dominance in the LSTM and CNN layers.

2.2 ARIMA

The Autoregressive Integrated Moving Average (ARIMA) model is a common statistical method used to study and predict time series data. A time series $\{Y_t\}$ is said to follow an ARIMA (p, d, q) model if the d[th] difference of of the series, denoted as $\{W_t\} = \nabla^d Y_t$ is a *autoregressive moving average* (ARMA) process. On the other hand, if $\{W_t\}$ follows an $ARMA$ model, then $\{Y_t\}$ is an $ARIMA(p,d,q)$ process, where p is the order of the *autoregressive* parameter, d is the order of *differencing*, and q is the order of the *moving average* [12]. The general mathematical formulation of the $ARIMA\ (p,\ d,\ q)$ process for the time series W_t is as follows:

$$W_t = \nabla^d Y_t.$$
$$W_t = \phi_1 W_{t-1} + \cdots + \phi_p W_{t-p} + e_t - \theta_1 e_{t-1} - \cdots - \theta_q e_{t-q} \quad (1)$$

Where W_t is the differenced series at time t, ϕ represents the autoregressive (AR) parameters, θ denotes the moving average (MA) parameters, and e_t is the random error value assumed to be white noise. The complexity of the model depends on three factors: p which is the order of the autoregressive part, d which is the order of differencing, and q which is the order of the moving average part.

By using $W_t = Y_t - Y_{t-1}$ and substituting it into the main equation, the model can be rewritten, as shown in (2) to relate the current observation directly to previous values:

$$Y_t = (1+\phi_1)Y_{t-1} + (\phi_2 - \phi_1)Y_{t-2} + \cdots + (\phi_p - \phi_{p-1})Y_{t-p} - \phi_p Y_{t-p-1} + e_t - \theta_1 e_{t-1} - \theta_2 e_{t-2} - \cdots - \theta_q e_{t-q} \quad (2)$$

2.3 LSTM

Long Short-Term memory is a development of RNN. LSTM uses memory cells to store long-term information, which is regulated by three gates: the forget gate, the input gate, and the output gate [13]. The forget gate removes irrelevant information, the input gate stores new information, and the output gate determines the output used for prediction. The advantage of LSTM lies in its ability to maintain gradients and prevent rapid gradient loss, which is a major problem in RNN [14].

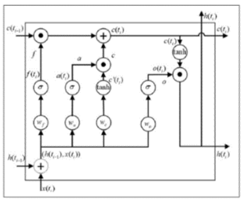


Figure 2. LSTM Network Layout

The picture above shows how a Long Short-Term Memory (LSTM) network is structured [15]. LSTM uses two types of activation function: *tanh* dan *sigmoid* (σ). Activation functions help neurons in the network process and pass information along, with each function taking in input and applying a particular mathematical rule [16]. Below is the mathematical formula for the Long Short-Term Memory model [15]:

$$a(t_i) = \sigma(w_a x(t_i) + w_{ha} h(t_{i-1}) + b_a) \quad (3)$$
$$f(t_i) = \sigma(w_f x(t_i) + w_{hf} h(t_{i-1}) + b_f) \quad (4)$$
$$c(t_i) = f_t \times c(t_{i-1}) + a_t \times \tanh\ (w_c x(t_i) + w_{hc}(h(t_{i-1}) + b_c) \quad (5)$$
$$o(t_i) = \sigma(w_o x(t_i) + w_{ho} h(t_{i-1}) + b_o) \quad (6)$$
$$h(t_i) = o(t_i) \times \tanh\ (c(t_i)) \quad (7)$$

where:

$x(t_i)$	: input value
$h(t_{i-1})$ dan $h(t_i)$	: output value at time t_{i-1} and t_i
$c(t_{i-1})$ dan $c(t_i)$	: cell state at time t_{i-1} and t_i

$b = \{b_a, b_f, b_c, b_o\}$: bias value of input gate, forget gate, internal state, and output gate
$\vec{W_1} = \{w_a, w_f, w_c, w_o\}$: weight values of input gate, forget gate internal state, and output gate.
$\vec{W_1} = \{w_{ha}, w_{hf}, w_{hc}, w_{ho}\}$: recurrent weight value.
$\vec{a} = \{a(t_i), f(t_i), c(t_i), o(t_i)\}$: output value of input gate, forget gate, internal state, and output gate.

The LSTM model was implemented to capture the non-linear residuals derived from the ARIMA process. The mathematical operations within the LSTM cell, which govern the flow of information through the forget, input, and output gates, are defined in equation 3 to equation 7. Then, to adapt these standard formulations to the specific requirements of palm oil proce forecasting, the following implementation strategies were applied:

a. Sliding Window and Lookback Period: the model processed the univariate series using a sliding window approach. The lookback period was determined through a series of empirical experiments testing various monthly intervals. The final window size was selected based on its ability to provide the most optimal error reduction during training, ensuring that the cell state $c(t_i)$ in equation 5 and the hidden state $h(t_i)$ in equation 7 effectively captured the localized temporal dependencies of the data.
b. Hyperparameter Optimization Strategy: a grid search was conducted to identify the best-performing architecture for the dataset. The search space included variations in the number of LSTM layers, hidden units, and learning rates. The final hyperparameters were chosen based on their collective ability to achieve the lowest Mean Squared Error (MSE) during the validation phase, ensuring the model's robustness against price volatility.
c. Training and convergernce: the model was trained using the ADAM optimizer, which is well-suited for non-stationary time series data. The training duration (epochs) was determined by monitoring the loss function to ensure the model reached a stable convergence point. This approach was taken to prevent overfitting, which is a significant risk when modelling univariate residuals.

2.4 CNN

Research on the use of CNNs for classification and time series has been growing. CNNs have two specialized layers: the convolutional layer and the pooling layer, which extract important features from the input data matrix. The CNN architecture for time series consists of several layers [17] :

a. Input layer: of size Nxk, where k is the number of univariate time-series data sets and N is the length of the data sets.
b. Convolutional layer: uses m filters of size lxk, with m and l adjustable as needed.
c. Pooling layer: can be either max pooling or average pooling.
d. Feature layer: combines the results of convolution and pooling into a single univariate time series for input to the fully connected layers.
e. Output layer: has n neurons, corresponding to the number of classes for time series classification. After the convolution layer, the resulting feature dimensionality is very large. To address this, a pooling layer is used to reduce the feature dimensionality and network training cost.

2.5 Hybrid ARIMA-LSTM

This approach deals with both the complex patterns of linear and non-linear structures in data, allowing for more accurate modeling. The hybrid model consists of two components: a linear component and a non-linear component, separated as follows [18] :

$$y_t = L_t + N_t \tag{8}$$

Where:
L_t : linear component at time t,
N_t : non-linear component at time t,
y_t : time series value at time t.

Forecasting using the hybrid ARIMA-LSTM method is performed in two steps. The first step uses the ARIMA model on the time series dataset to obtain the forecast results for the linear component using equation 2. The second step trains the LSTM model on the residuals from the ARIMA model. The residual value can be obtained using the following equation [18] :

$$R_t = y_t - \hat{L}_t \tag{9}$$

Where:
R_t : residual value at time t,
y_t : actual value of the time series at time t,
$\hat{L}_t$: forecasted value of the time series at time t.

The residual is an important measure of model fit, measuring how far the predicted value is from the actual value at time t. by using the residuals from a linear model as training data for the LSTM, the LSTM can focus on non-linear patterns because linear patterns are already handled by ARIMA. The final predictions of the hybrid model is obtained by summing the prediction results from the ARIMA and LSTM models using the following equation [18] :

$$\hat{y}_t = \hat{L}_t + \hat{N}_t \tag{10}$$

Where:

$\hat{y}_t$: the forecasted value at time t,
$\hat{L}_t$: the forecasted value of the ARIMA model,
$\hat{N}_t$: the residual forecasted value of the LSTM model at time t.

2.6 Hybrid CNN-LSTM

The CNN-LSTM hybrid method integrates the CNN and LSTM algorithms. CNN focuses on important features in the data, while LSTM retains temporal information. By combining these two methods, the hybrid approach leverages the strengths of both [19]. The architecture of the CNN-LSTM hybrid method is depicted in the following scheme.

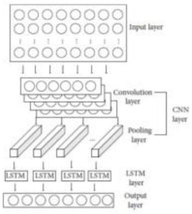


Figure 3. Hybrid CNN-LSTM Architecture

The architecture illustrates that the CNN-LSTM hybrid approach integrates the convolutional layer from CNN with the LSTM network for prediction purposes [20]. The CNN-LSTM model is constructed by inserting the LSTM algorithm at the final layer of the CNN structure. The input vector is first processed through the CNN algorithm before being passed to the LSTM network. During the CNN phase, processes: convolution and pooling. Once these steps are completed, the data is sent to the LSTM layer for forecasting [9]. The data processing flow is described as follows:

a. CNN Phase: the input vector is first processed through convolutional layers to perform non-linear feature extraction, followed by pooling layers to reduce dimensionality and network training costs.
b. LSTM Phase: the extracted spatial features are then passed to the LSTM layers, where the model learns sequential patterns and trends over time.
c. Output layers (Forecasting Modification): to adapt the architecture for a forecasting task rather than classification, the final layers are specifically modified. Unlike classification models that use multiple neurons and a softmax activation function to predict class probabilities, this model employs a single Dense layer with one neuron and a linear activation function. This configuration allows the network to map the hierarchical spatial-temporal features directly into a single continuous value, representing the final predicted price.

2.7 Model Evaluation

One of the steps to measure the performance of a built model is model evaluation. Many error measures can be used, one of which is MAPE (Mean Absolute Percentage Error), which can be calculated using the following equation [21] :

$$MAPE = \frac{100\%}{N} \sum_{t=1}^{n} \left| \frac{A_t - F_t}{A_t} \right| \tag{11}$$

In the above equation, n is the number of learning samples, A_t is the actual value, and F_t is the estimated value.

3. RESULT AND ANALYSIS

3.1 Hybrid ARIMA-LSTM Model Development

The Crude Palm Oil (CPO) price data from 1993 to 2024 has been very unpredictable and has had major changes in its pattern because of big events around the world. In the past, the market was affected by the Asian Financial Crisis in 1997-1998, which made the prices inside and outside the region different because of the falling value of money [22]. Then, the 2008 Global Financial Crisis brought huge price swings. Prices initially surged due to the growing global demand for biofuels which linked vegetable oil markets to energy markets [23]. However, this trend reversed sharply in the second half of the year, where prices dropped by more than 60% as the financial crisis triggered a massive contraction in global demand [24]. In 2022, the data exhibited extreme volatility and non-stationarity, driven by two major exogenous shocks. First, the Russia-Ukraine conflict severely disrupted the global supply of sunflower oil, triggering a substitution effect towards palm oil [25]. Second, this volatility was exacerbated by Indonesia's temporary export ban in April 2022, which abruptly removed the world's largest supplier from the market [26]. Subsequently, in the 2023 – 2024 period, the implementation of Indonesia's B35 biodiesel mandate has established a new structural demand, effectively creating a higher price floor and altering the long-term average of the series [27]. These structural breaks justify the necessity of the rigorous data transformation and stationarity checks performed in this research.

The stationarity of the data on the mean aspect was visually examined using the autocorrelation (ACF) graph in Figure 4. There is a consistent fluctuating trend indicating that the data is not yet stationary in the mean. This is supported by the ACF graph which shows a slow decay pattern from lag to lag, a characteristic of non-stationary data. The ADF test supports this finding, with a p-value of 0.108, which is higher than the 5% significance level. That means we can't confidently say the data is stationary. Next, we checked the stationarity of the variance and found that it is also non-stationary. To make the data stationary in both the mean and variance, we applied a Box-Cox transformation and then did one differencing. The results are shown in Figure 5, where the data starts to move around a steady mean value. After the transformation, the ADF test gave a p-value of 0.000147, which is below the 5% significance level. This means the data now meets the stationarity assumption. In this context, the stationarity assumption. In this case, the stationarity assumption means the time series must meet three main conditions: (1) a constant mean that does not change with time, (2) a constant variance across all time intervals, and (3) an autocovariance structure that depends only on the lag between observations rather than the actual time of recording. Meeting these conditions helps the ARIMA model work well by ensuring it can accurately capture linear patterns without being affected by trends or sudden changes.

To deal with concerns regarding the statistical significance of the model's performance and to ensure the results were not an artifact of a specific data split, this research implemented a time series split k-fold cross validation with k = 3 only on the training data. This approach functioned as a series of repeated runs, where the model was iteratively trained and validated across different temporal segments of the price history.

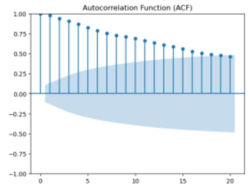


Figure 4. ACF Plot of Training Data

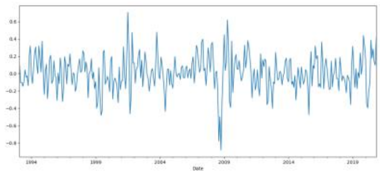


Figure 5. Plot of Training Data After Transformation & Differentiating

The stationary data goes through a model identification process. To identify the ARIMA model, ACF and PACF plots are used. By looking at the results from these plots, several possible models were chosen. These models include: ARIMA (0,1,1), ARIMA (1,1,0), ARIMA (1,1,1), ARIMA (1,1,2), ARIMA (2,1,1), ARIMA (2,1,2), ARIMA (1,1,4), ARIMA (2,1,4).

Table 1. Model Identification and Tentative Model Parameter Estimation

Model	Parameter Significance	AIC	BIC
ARIMA (0,1,1)	All parameters are significant	3416.133758	3423.756040
ARIMA (1,1,0)	All parameters are significant	3423.337697	3430.959979
ARIMA (1,1,1)	Some parameters are significant	3417.404329	3428.837752
ARIMA (1,1,2)	All parameters are significant	3415.984248	3431.228812
ARIMA (2,1,1)	Some parameters are significant	3419.063236	3434.307800
ARIMA (2,1,2)	All parameters are significant	3415.456407	3434.512112
ARIMA (1,1,4)	Some parameters are significant	3411.237335	3434.104181
ARIMA (2,1,4)	All parameters are significant	3407.892560	3434.570547

The parameter estimation results in the table indicate that the ARIMA (2,1,4) model has the lowest AIC value but includes insignificant parameters. On the other hand, the ARIMA (0,1,1) model provides the lowest BIC value with all parameters being statistically significant. Considering both criteria, the ARIMA (0,1,1) model was selected as the best model because the model with the lowest AIC value did not meet the parameter significance requirement.

3.1.1 Model Diagnostic Test ARIMA

Diagnostic test on the model residuals was conducted exploratively and formally using the Residual Diagnostic Panel shown in figure 6. The exploratory results from the Normal Q-Q plot in Figure 6 show a quantile plot that shows a distribution pattern of points that does not follow the 45° diagonal line, indicating that the residuals tend not to be normally distributed. These results are reinforced through formal testing with the Kolmogorov-Smirnov test, which produces a p-value of 3.424×10^{-14}, which is smaller than 0.05. So, the assumption of normality is rejected. However, this violation isn't considered serious because the data set is quite large [28].

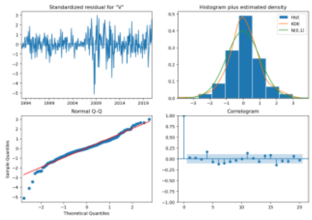


Figure 6. Residual Diagnostic Panel

Regarding independence, the Correlogram in Figure 6 shows a significant lag, indicating the data is not white noise. However, the formal Ljung-Box test yields a p-value 0f 0.4642, which is greater than 0.05 indicating insufficient evidence to reject the null hypothesis that the residuals are uncorrelated (independent).

Finally, the Standardized Residual Plot in Figure 6 was used to asses variance homogeneity. The graph shows a relatively uniform residual bandwidth, but a strong outlier in the middle of the graph causes the residual bandwidth to tend to be unequal, indicating that the residual variance is not homogeneous. These results are supported by the Breusch-Pagan test, with a p-value < 0.05 at 1.112×10^{-5}, indicating that the variance homogeneity assumption is not met, indicating the need for a more advanced modelling approach to accommodate this. Problems such as random fluctuations in residuals and the issue of vanishing gradients due to long-term dependencies, which pose obstacles, can be addressed with the LSTM approach and chosen as a more adaptive solution to complex residual characteristics [29].

3.1.2 LSTM Model Development

An LSTM network model was built using Tensorflow to model residuals, which are the differences between the actual and predicted values from the ARIMA model that contain non-linear patterns not captured by ARIMA, with the aim of improving overall forecasting accuracy. These residual values were normalized to a range of 0 - 1 to be suitable for LSTM processing, so can used for training and making predictions with the best set of parameters.

The process of finding the best parameters included using time series split k-fold cross-validation (with k=3) to ensure the data's time-based order was kept intact. This split was implemented using the time series split module from the scikit-learn library, which follows an expanding window strategy to maintain the temporal dependency of the data. In this scheme, the training set is partitioned into $k + 1$ consecutive segments; in the first fold, the model is trained on the first segment and validated on the second. In each subsequent fold, the training window expands to include all previous segments, while the immediately following segment is used for validation. This ensures that the model is always evaluated on a future validation set that chronologically follows the training data, thereby preventing data leakage and ensuring the reproducibility of the results:

Table 2. LSTM Model Architecture Design

Hyperparameter	Value
Number of layers	1
Lookback	[6, 12, 24]
Optimizer	Adam
Number of neurons	[4, 20, 40, 50, 60, 80, 100, 120, 150, 200]
Learning rate	[0.001, 0.0001, 0.00001]
Batch Size	[16, 32]
Epochs	[10, 50, 100, 150, 200]

It is important to clarify that this cross-validation strategy was employed strictly during the hyperparameter tuning phase to ensure model robustness and to find the optimal configuration, a grid search was conducted, evaluating a total of 900 unique combinations based on the parameters predefined in Table 2. The search was performed on a MacBook Air with an Apple M1 chip, utilizing its high-performance CPU cores to ensure computational efficiency and convergence within a reasonable timeframe.

However, for the final comparative evaluation, a sequential single split (hold out) strategy was adopted to simulate a real-world forecasting scenario where the model predicts the unknown future based solely on historical data, thereby preventing look ahead bias.

Table 3. Best Combination

Number Of Layers	Lookback	Optimizer	Number Of Neurons	Learning Rate	Batch Size	Epochs
1	6	Adam	4	0.001	16	50

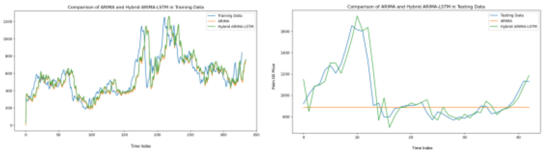
3.1.3 Hybrid ARIMA-LSTM Results

The hybrid method was developed by combining ARIMA and LSTM prediction results on the residuals according to equation 10.

Table 4. Comparison of Evaluation Values of Training Data and Testing Data

Methods	Training Data		Testing Data	
	RMSE	MAPE	RMSE	MAPE
ARIMA (0,1,1)	44.0181	5.49%	253.6895	13.66%
Hybrid ARIMA-LSTM	43.0568	5.37%	104.7583	7.38%

The results show that the hybrid ARIMA-LSTM method outperforms the ARIMA method for Table 4, with lower MAPE and RMSE values on both training and test data, as also visually indicated in Figure 7.



(a) (b)

Figure 7. (a) Training Data Forecast Plot (b) Testing Data Forecast Plot

Figure 7 shows that the ARIMA and hybrid ARIMA-LSTM methods produce different forecast patterns on the train and test data. The ARIMA method is less capable of capturing patterns in the test data due to its limitations in modeling non-linear and complex data, while the hybrid ARIMA-LSTM method demonstrates more accurate results due to LSTM's ability to memorize historical information and recognize long-term patterns without requiring stationary data.

Regarding the statistical significance of the results, standard confidence intervals (CI) were not computed for the hybrid ARIMA-LSTM model due to theoretical constraints. The hybrid forecast combines a linear component (ARIMA) and a non-linear residual component (LSTM). While ARIMA residuals typically assume a Gaussian distribution, the error distribution of the LSTM component is non-parametric and analytically intractable. Applying standard confidence intervals based on normality assumptions would be methodologically invalid in this hybrid context because the resulting combined residuals likely exhibit skewness or heavy tails typical of high-volatility data. Therefore, this study adopts a robust validation approach by prioritizing consistent error reduction (indicated by RMSE and MAPE) on out-of-sample data. This strategy focuses on point-prediction accuracy, providing a realistic assessment of the model's forecasting capability without relying on violated statistical assumptions.

3.2 Hybrid CNN-LSTM Model Development

3.2.1 CNN-LSTM Modelling: Preprocessing and Architecture

The modelling data used is the same as in the previous model, which is a univariate time series consisting of monthly palm oil prices. In the preprocessing stage, the data that has been confirmed to have no missing values or duplications is the normalized using Min-Max Scaling to the range [0,1] and transformed into a supervised learning format using a sliding window of 24 timesteps then reshaped into a 3D form (samples, timesteps, features).

The model is designed by combining a CNN layer for extracting spatial features and an LSTM network layer for learning temporal patterns. To ensure the optimization process was through and fair, hyperparameter tuning was conducted on both layers by evaluating different paremeter combinations. This tuning used time series split k-fold cross-validation with 3-fold, and it was only done on the training data. By using this rolling validation method, the model's parameter was adjusted without using the 13% of data that was kept aside as a test set. This helped keep the evaluation truly out of sample and avoided any data leakage. After tuning, the model was set up using the ADAM optimizer and the Mean Squared Error (MSE) loss function. Both MSE and MAE were used as ways to measure how well the model was performing. The full design of the model and the extract range of parameters searched are shown in Table 5.

Table 5. CNN-LSTM Model Architecture Design

Hyperparameter	Value
Number of layers	1
Lookback	[6, 12, 24]
Optimizer	Adam
LSTM Units	[50, 100, 120, 150, 200]
Learning rate	[0.001, 0.0001, 0.00001]
Batch Size	[16, 32]
Epochs	[10, 50, 100, 150, 200]
Kernel Size	1
Filters Size	[20, 32, 64]
Conv Layer Num	[1, 5]
Pool Size	[1, 2]
Activation	Relu, sigmoid

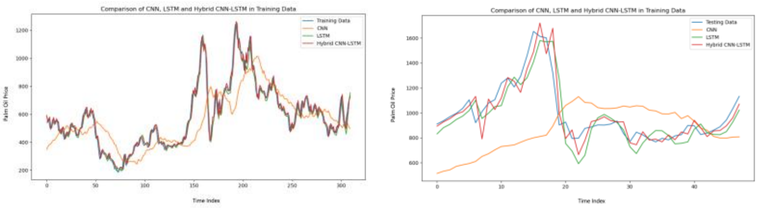
The best combination is selected based on the lowest evaluation metric value which will then be used to rebuild, train the model, and test the model.

Table 6. Best CNN-LSTM Combinations

Hyperparameter	Value
Number of layers	1
Lookback	24
Optimizer	Adam
LSTM Units	200
Learning rate	0.0001
Batch Size	16
Epochs	100
Kernel Size	1
Filters Size	20
Conv Layer Num	1
Pool Size	1
Activation	Relu

3.2.2 Hybrid CNN-LSTM Model Results and Evaluation

The completed model training was then saved in HDF5 format using the Keras API from the TensorFlow library to enable reuse without the need for retraining. One way to evaluate the performance of the hybrid CNN-LSTM model was to visually compare the prediction results with the actual data, both for the training and test sets, by looking at a graph, as shown in the figure below:



(a) (b)

Figure 8. (a) CNN-LSTM Training Data Forecast Plot (b) CNN-LSTM Test Data Forecast Plot

The evaluation results show that the hybrid CNN-LSTM model developed is able to provide the best performance in capturing trend patterns and price fluctuations, where the model produces precises predictions with the smallest deviation on training data, and on test data the hybrid CNN-LSTM model maintains its accuracy in following the actual data pattern. In comparison, the individual CNN model shows a larger deviation value, especially in capturing sharp fluctuation patterns. So overall the hybrid CNN-LSTM model has superior adaptability in modelling the complexity of palm oil price time series data compared to the individual model approach.

3.3 Determining the Best Model

The selection of the optimal forecasting model is grounded in a comparative analysis of predictive performance on out-of-sample test data. Given the high volatility of the dataset, the primary criterion for determining the best model is the minimization of deterministic error metrics, specifically Mean Absolute Percentage Error (MAPE) and Root Mean Square Error (RMSE).

Table 7. Performance Evaluation of Forecasting Methods on Training and Test Data

Methods	Training Data		Test Data	
	RMSE	MAPE	RMSE	MAPE
ARIMA (0,1,1)	44.0181	5.49%	253.6895	13.66%
Hybrid ARIMA-LSTM	43.9284	5.91%	106.3490	7.38%
CNN	166.6746	21.63%	365.3381	28.59%
LSTM	42.4691	5.59%	118.6301	9.48%
Hybrid CNN-LSTM	38.0583	5.26%	96.0624	6.69%

Based on the empirical results presented in Table 7, the hybrid CNN-LSTM is identified as the superior model. It achieved the lowest error rates among all tested architectures, recording a MAPE of 6.69% and an RMSE of 96.0624. This performance outperforms both the single-method models and the hybrid ARIMA-LSTM with MAPE of 7.38%. The selection of CNN-LSTM is justified not merely by the numerical reduction in error, but by its architectural capability. Consequently, the hybrid CNN-LSTM provides the strongest empirical evidence of predictive accuracy and is selected as the best model for this research.

4. CONCLUSION

This research evaluated the performance of hybrid forecasting models in predicting palm oil prices high volatility. The empirical results demonstrate that the CNN-LSTM model achieved the most accurate predictions, recording the lowest error rates on the test data (MAPE of 6.69% and RMSE of 96.0624) compared to the Hybrid ARIMA-LSTM (MAPE of 7.38%) and single-method models. This finding indicates that the CNN-LSTM architecture which combines feature extraction with temporal learning offers a superior mechanism for filtering noise and capturing non-linear market flustuations compared to traditional or simpler hybrid approaches.

A limitation of this study is the absence of analytical confidence intervals for the hybrid and deep learning models, which is often attributed to their non-parametric and black-box nature. While this research ensures reliability through deterministic metrics (RMSE and MAPE), future work could enhance uncertainty quantification using boostrap based estimation or Bayesian neural networks to provide probabilistic intervals without relying on Gaussian assumptions. Furthermore, a formal statistical comparison such as the Diebold-Mariano test was not implemented due to the relatively limited size of the out-of-sample testing set, which may constrain the statistical power required to draw stable conclusions. Although the current evaluation focused on architectural feasibility and error minimization, future research could further extend this work by applying rigorous hypothesis testing across boarder datasets to strictly validate the statistical significance of the performance differences.

Beyong the numerical comparison, the enhanced accuracy of the hybrid model serves as a vital tool for industry practitioners in risk management and for policymakers in formulating proactive land-use strategies to mitigate ecological risks driven by market volatility. These findings provide a scalable framework for forecasting other volatile agricultural commodities beyond the palm oil industry.

5. REFERENCES

[1] D. C. Montgomery, C. L. Jennings, and M. Kulahci, *Introduction to Time Series Analysis and Forecasting*, 2nd ed. Hoboken, NJ: Wiley, 2015.

[2] K. He, L. Ji, C. W. D. Wu, and K. F. G. Tso, "Using SARIMA–CNN–LSTM approach to forecast daily tourism demand," *J. Hosp. Tour. Manag.*, vol. 49, no. October 2020, pp. 25–33, 2021, doi: 10.1016/j.jhtm.2021.08.022.

[3] R. Fazira, D. Yudistira, and L. Sofinah Harahap, "Evaluasi Kinerja Model RNN & LSTM untuk Prediksi Magnitude Gempa di Indonesia," *Mars J. Tek. Mesin, Ind. Elektro Dan Ilmu Komput.*, vol. 2, no. 6, pp. 62–75, 2024, doi: 10.61132/mars.v2i6.498.

[4] S. Sen, D. Sugiarto, and A. Rochman, "Komparasi Metode Multilayer Perceptron (MLP) dan Long Short Term Memory (LSTM) dalam Peramalan Harga Beras," *Ultim. J. Tek. Inform.*, vol. 12, no. 1, pp. 35–41, 2020, [Online]. Available: https://ejournals.umn.ac.id/index.php/TI/article/view/1572/954

[5] C. Alkahfi, A. Kurnia, and A. Saefuddin, "Perbandingan Kinerja Model Berbasis RNN pada Peramalan Data Ekonomi dan Keuangan Indonesia," *MALCOM Indones. J. Mach. Learn. Comput. Sci.*, vol. 4, no.4, no. October, pp. 1235–1243, 2024.

[6] Y. A. Nugroho and H. A. Hutahaean, "Pengembangan Model Deep Learning LSTM dan CNN untuk Peramalan Penjualan Sepeda Motor di Indonesia," *Jupiter Publ. Ilmu Keteknikan Ind. Tek. Elektro dan Inform.*, vol. 3, no. 2, pp. 94–104, 2025, doi: 10.61132/jupiter.v3i2.795.

[7] M. Qaim, K. T.Sibhatu, H. Siregar, and I. Grass, "Environmental , Economic , and Social Consequences of the Oil Palm Boom," *Annu. Rev. Resourse Econ.*, vol. 12, no. 1, pp. 321–344, 2020, doi: 10.1146/annurev-resource-110119-024922.

[8] G. P. Zhang, "Time series forecasting using a hybrid ARIMA and neural network model," *Neurocomputing*, vol. 50, pp. 159–175, doi: 10.1016/S0925-2312(01)00702-0.

[9] W. Lu, J. Li, Y. Li, A. Sun, and J. Wang, "A CNN-LSTM-based model to forecast stock prices," *Complexity*, vol. 2020, p. 6622927, 2020, doi: 10.1155/2020/6622927.

[10] M. Iaousse, Y. Jouilil, M. Bouincha, and D. Mentagui, "A Comparative Simulation Study of Classical and Machine Learning Techniques for Forecasting Time Series Data," *Int. J. Online Biomed. Eng.*, vol. 19, no. 8, pp. 56–65, 2023, doi: 10.3991/ijoe.v19i08.39853.

[11] E. K. M. Uskono, "Aplikasi Metode ARIMA, LSTM, dan Hybrid ARIMA-LSTM pada Peramalan Harga Crude Palm Oil (CPO) Dunia," Institut Pertanian Bogor, 2023.

[12] J. D. Cryer and K.-S. Chan, *Time Series Analysis with Application in R*, 2nd ed. New York: Springer New York, 2008.

[13] X. Li, D. Li, Y. Cheng, and W. Li, "Forecasting the Volatility of Educational Firms Based on HAR Model and LSTM Models Considering Sentiment and Educational Policy," *Heliyon*, vol. 10, no. 19, 2024, doi: https://doi.org/10.1016/j.heliyon.2024.e38560.

[14] A. Agga, S. A. Abbou, Y. El Houm, and M. Labbadi, "Short-Term Load Forecasting Based on CNN and LSTM Deep Neural Networks," *IFAC-PapersOnLine*, vol. 55, no. 12, pp. 777–781, 2022, doi: 10.1016/j.ifacol.2022.07.407.

[15] H. Abbasimehr, M. Shabani, and M. Yousefi, "An optimized model using LSTM network for demand forecasting," *Comput. Ind. Eng.*, vol. 143, no. July 2019, p. 106435, 2020, doi: 10.1016/j.cie.2020.106435.

[16] A. Z. A. Ra, "COMPARISON OF DYNAMIC FACTOR MODELS (DFM) AND LONG SHORT-TERM MEMORY (LSTM) NETWORKS IN FORECASTING HOUSEHOLD CONSUMPTION," Institut Pertanian Bogor, 2022.

[17] B. Zhao, H. Lu, S. Chen, J. Liu, and D. Wu, "Convolutional neural networks for time series classification," *J. Syst. Eng. Electron.*, vol. 28, no. 1, pp. 162–169, 2017, doi: 10.21629/JSEE.2017.01.18.

[18] M. Lenderink, "Accounts Receivable Cash Flow Forecasting: A comparison between SES, ARIMA, LSTM and Hybrid ARIMA-LSTM," Tilburg University, 2022.

[19] Y. Widhiyasana, T. Semiawan, I. G. A. Mudzakir, and M. R. Noor, "Penerapan Convolutional Long Short-Term Memory untuk Klasifikasi Teks Berita Bahasa Indonesia," *J. Nas. Tek. Elektro dan Teknol. Inf.*, vol. 10, no. 4, pp. 354–361, 2021, doi: 10.22146/jnteti.v10i4.2438.

[20] J. Nathaniel and V. J. L. Engel, "Penerapan Nonpooling CNN-LSTM Untuk Prediksi Pemakaian Obat Rumah Sakit," Institut Teknologi Harapan Bangsa, 2022. [Online]. Available: https://repository.ithb.ac.id/id/eprint/30/9/1117012_Paper-TA.pdf

[21] D. Al Mahkya, K. A. Notodiputro, and B. Sartono, "Extra Trees Method for Stock Price Forecasting With Rolling Origin Accuracy Evaluation," *Media Stat.*, vol. 15, no. 1, pp. 36–47, 2022, doi: 10.14710/medstat.15.1.36-47.

[22] F. Bresciani, G. Feder, D. O. Gilligan, H. G. Jacoby, T. Onchan, and J. Quizon, "Weathering the Storm : The Impact of the East Asian Crisis on Farm Households in Indonesia and Thailand," *World Bank Res. Obs.*, vol. 17, no. 1, pp. 1–20, 2002.

[23] D. Mitchell, "A Note on Rising Food Prices," Washington, D.C, 2008.

[24] J. Baffes and T. Haniotis, "Placing the 2006 / 08 Commodity Price Boom into Perspective," Washington, D.C, 2010.

[25] World Bank, "Commodity Markets Outlook: The Impact of the War in Ukraine on Commodity Markets," Washington, D.C, 2022.

[26] J. Glauber, D. Labor, and A. Mamun, "Food export restrictions have eased as the Russia-Ukraine war continues, but concerns remain for key commodities," IFPRI Blog. [Online]. Available: https://www.ifpri.org/blog/food-export-restrictions-have-eased-russia-ukraine-war-continues-concerns-remain-key/

[27] S. G. C. Insights, "Global Biofuels Special Report: 5 issues to watch in 2025 and what you can expect," New York, NY (Markas besar S&P Global), 2025.

[28] K. A. Notodiputro and Y. Anggraini, *R, Data dan Statistik Deskriptif.* Bogor: IPB Publisher, 2023.

[29] E. Arif, E. Herlinawati, D. Devianto, M. Yollanda, and D. Permana, "Hybridization of long short-term memory neural network in fractional time series modeling of inflation," *Front. Big Data*, vol. 6, 2024, doi: 10.3389/fdata.2023.1282541.