IJSA (2024), Vol. 8, No. 2, 95-111
DOI: https://doi.org/10.29244/ijsa.v8i2p95-111

Indonesian Journal of Statistics and Its Applications
Available at https://journal-stats.ipb.ac.id/index.php/ijsa
e-ISSN: 2599-0802

Comparison Between SARIMA and DeepAR with Optuna Hyperparameter Optimization for Estimating Rice Production Data in Indonesia*

Muhammad Farhan Zahid[1‡], Anwar Fitrianto[2], Pika Silvianti[3], and Aam Alamudi[4]

[1,2,3,4]Department of Statistics, IPB University, Indonesia
‡corresponding author: farhan_zahid_muhammad@apps.ipb.ac.id

Abstract

A forecast predicts future events that have significantly impacted our society, especially when facing time-sensitive issues like food availability. Food was critical to ensuring people's welfare, especially in a country like Indonesia, which has a large population. Availability and access to rice are a vital need for the people of Indonesia. Rice is not only the primary source of carbohydrates but also has a central role in Indonesian society's cultural and social aspects. Forecasting can be a strategy to anticipate fluctuations in food demand and supply, serving as an important instrument for the government sand stakeholders to make effective decisions. The growing period of rice, which is heavily influenced by seasonality, makes DeepAR and SARIMA techniques an excellent solution to this problem. Both methods offered the ability to address features in rice production such as trends, seasonality, and anomaly effects. This study aimed to compare the performance of the machine learning method, DeepAR, and the classic forecasting method, SARIMA, in estimating seasonal data pattern, rice yield predictions. This study demonstrates that DeepAR, especially when optimized with Optuna, outperforms SARIMA in forecasting rice production in Indonesia, as evidenced by superior performance in key evaluation metrics such as root mean square error (RMSE) and mean absolute percentage error (MAPE).

Keywords: Comparison, DeepAR, Optuna, Rice Production, SARIMA

* Received: Jun 2024; Reviewed: Jul 2024; Published: Dec 2024

1. Background

A forecast is a prediction of some future event(s) (Montgomery *et al.,* 2015), which has taken a significant role in our society, aiding humans in making critical decisions, particularly when facing time-sensitive issues like weather, temperature, crop yield, and food availability. Fattah *et al.,* (2018) emphasized that forecasting is crucial for inventory management, which can be applied to many aspects of human life. Sumarsono *et al.,* (2019) further stated that the need for food is a human right and one of the basic human needs. The issue of food is a key challenge faced by every country. The failure of a government to provide adequate food for its citizens can result in serious impacts, such as a slowdown in production and incidents of mass death as happened in the Irish famine (Forsberg, 2018). Therefore, the use of forecasting is required to predict days of both excess and insufficient production in order to design an adequate policy for maintaining food security.

In Indonesia, rice is the staple food for the population. As a tropical crop, rice thrives with 1500-2000 mm of annual rainfall. According to Patria *et al.,* (2017), changes in rainfall patterns and temperature increases greatly affect rice production because rice growth in Indonesia, mostly dominated by traditional planting methods, is highly dependent on the rainy season and dry season cycles. Rahim *et al.,* (2017) also mentioned that rice grows within 100-115 days or about 3-4 months, which usually falls between October and March. The research is also supported by BPS data which states that rice yields peak in March. Additionally, Ohyver and Pudjihastuti (2018) asserted that various forecasting methods can be applied to address this issue.

The autoregressive integrated moving average (ARIMA) method is recognized as a powerful and flexible tool for performing time series analysis and forecasting (Montgomery *et al.,* 2015). However, while ARIMA is effective, natural phenomena often depend on past values at multiples of several lags, especially when related to calendar years. This necessitates modifications to ARIMA to accommodate seasonality, leading to the development of seasonal autoregressive integrated moving average (SARIMA) (Shumway *et al.,* 2011).

However, ARIMA cannot work well with small dataset leading to the use of neural network method. Hardware development makes the use of machine learning such as neural networks increasingly developed. Amazon research team created a method called DeepAR. Salinas *et al.,* (2020) stated that DeepAR has several advantages over classic forecasting methods, one of which is its ability to learn from several similar time series, allowing it to provide accurate estimates even from very small datasets.

This study aims to compare the performance of the machine learning method, DeepAR, and the classic forecasting method, SARIMA, in predicting rice yield. Given the seasonal nature of the data, opting for the regular ARIMA is avoided, as SARIMA is better suited for handling seasonal patterns.

2. Methodology

2.1 Data

The data used in this study is monthly rice production in Indonesia from January

2018 to December 2022. Rice production is calculated based on the product of the harvested area and field productivity. The data is obtained from the book "Luas Panen dan Produksi Beras" 2018-2022, available for download on the website https://www.bps.go.id/

2.2 Rice Growth

Food security stands as a paramount concern in Indonesia. In 2020, the Ministry of Agriculture devised a strategic approach aimed at augmenting production capacity and fortifying the rice reserves of the nation. The productivity of rice, a staple crop in Indonesia, has exhibited a tendency toward stagnation, experiencing a modest annual increase of 0.24%, whereas the expansion in cultivated paddy fields has seen a more substantial decrease rate of -0.71% annually (Be, 2022).

Preserving community food security mandates, the establishment of a comprehensive national food reserve—a repository of food stocks strategically distributed across Indonesia. This reservoir serves the dual purpose of meeting consumption needs and addressing challenges related to food scarcity, supply disruptions, and price volatility.

Rice, being a pivotal food crop extensively cultivated by Indonesian farmers, holds a primary status in the dietary habits of Indonesians. The cultivation cycle of rice spans approximately 100-115 days, translating to a duration of 3-4 months (Rahim *et al.,* 2017). Shifts in precipitation patterns and escalating temperatures have a profound impact on rice production, with the temporal cycle of rice plants lengthening in direct correlation to elevated altitudes and temperatures. Furthermore, whether grown in rain-fed or irrigated paddy fields, rice yields are anticipated to decline in tandem with rising temperatures (Patria *et al.,* 2017).

2.3 SARIMA

2.3.1. General Theory and Equation

Some events or data sometimes have seasonal patterns such as crop yield which is very dependent on cycle of rainfall and temperature (Bang *et al.,* 2019) or sales volume of agricultural products due to demand on some specific season (Yoo and Oh, 2020). Seasonal autoregressive integrated moving average (SARIMA) can be a solution in such cases. According to Cryer and Chan (2008), the SARIMA model is denoted by ARIMA(p,d,q)x(P, D,Q)$_S$. The SARIMA model can also be written as

$$\phi_p(\mathrm{B})\Phi_P(\mathrm{B}^S)(1-\mathrm{B})^d(1-\mathrm{B}^S)^D \mathrm{y}_t = \theta_q(\mathrm{B})\Theta_Q(\mathrm{B}^S)\varepsilon_t \quad (1)$$

where,

p, d, and q : non-seasonal order AR, differencing, MA
P, D, and Q : seasonal order AR, differencing, MA
y_t : data in period t
ε_t : residual in period t
B : backward shift
S : Seasonal Order

2.3.2. Augmented Dicky-Fuller

A time series is said to be stationary if its properties are not affected by a change in the time origin (Montgomery *et al.,* 2015). Augmented Dickey-Fuller Test is used to test whether the series has a unit root. If the series does not have a unit root then the series is stationary (Halim and Bisono, 2008). If non-stationarity is suspected, differencing should be considered (Montgomery *et al.* 2015). Hypothesis for ADF Test are as follow:

H_0 : The Data is not stationary

H_1 : The Data is stationary

The formula to compute ADF test statistics is

$$t_{hit} = \frac{\hat{\phi}}{SE(\hat{\phi})} \tag{2}$$

where $SE(\hat{\phi})$is the standard error of the least squares estimate of $\hat{\phi}$. If the calculated t_{hit} is greater than t_{table} or the p-value is smaller than the significant level α, then reject H_0.

2.3.3. BocCox test

Stationarity serves as a pivotal concept, comprising two essential facets: stationarity in mean and stationarity in variance. Stationarity in mean entails the constancy of the mean value of a time series over time, indicating a consistent average behavior. Conversely, stationarity in variance reflects the stability of the variability or dispersion of data points around the mean. BoxCox transformation was used to handle the non-stationarity in variance. The Box-Cox transformation emerges as a powerful technique to tackle this challenge by stabilizing the variance of a dataset through a power transformation. Mathematically, the transformation is defined by Box and Cox (1964) as follow:

$$y_t(\lambda) = \begin{cases} \dfrac{{y_t}^{\lambda} - 1}{\lambda}, & \lambda \neq 0 \\ \ln y_t, & \lambda = 0 \end{cases} \tag{3}$$

where y_t denotes series, λ denotes transformation parameter that must be in $[0,1]$ interval.

2.3.4. Autocorrelation Function

The autocorrelation function (ACF) a statistical method used to identify the presence of correlation between a time series and its own lagged values. Autocorrelation measures how a series is correlated with its own past values at different time lags. The autocorrelation function is commonly denoted by ρ_k where ρ_k is the autoregressive coefficient of lag k with k = 0,1,2,…, k. The formula for the sample autocorrelation at lag *k* is given by:

$$\rho_k = \frac{E[(y_t - \mu)(y_{t-k} - \mu)]}{E[(y_t - \mu)^2]E[(y_{t-k} - \mu)^2]} \tag{4}$$

where n denotes amount of data and y_t denotes is the value of the time series at time.

2.3.5. Partial Autocorrelation Function

Partial autocorrelation function (PACF) is used in time series analysis to measure the correlation between observations of a time series that are separated by a specified number of time steps, while controlling for the values of the observations at intermediate lags. PACF is a function to that behaves like ACF of MA models, but for AR models (Shumway *et al.,* 2011).

$$\alpha_{kk} = \frac{\rho_k - \sum_{j=1}^{k-1} \alpha_{k-1,j}\, \rho_{k-j}}{1 - \sum_{j=1}^{k-1} \alpha_{k-1,j}\, \rho_j} \quad (5)$$

2.3.6. Akaike's Information Criterion (AIC)

Akaike's information criterion was the first model selection criterion to gain widespread attention in the statistical community, and continues to be one of the most widely known and used model selection tools in statistical practice. (Cavanaugh, 2019). The formula of AIC as stated by Akaike itself (1974):

$$AIC = -2\ln(\text{maximum likelihood}) + 2k \quad (6)$$

where k denotes the number of independent parameters that are fitted for the model being assessed.

However, a correction term intended to adjust the bias for a small sample size needs to be added. The AIC corrected for a small sample size needs to be added. The AIC corrected for small sample bias (AIC_C) is defined by (Sugiura, 1978) as:

$$\text{AIC}_C = AIC + \frac{2k(k+1)}{n-k-1} \quad (7)$$

where, n is the sample size, and k an AIC are defined above

AIC_C is more generally used in place of AIC. Lower AIC_C (or AIC) value indicates a better trade-off between goodness of fit and model complexity. Consequently, the model with the lowest AIC_C (or AIC) was selected among competing models.

2.4 DeepAR
2.4.1. General Theory and Equation

DeepAR is a supervised forecasting technique for forecasting scalar time series using recurrent neural networks (RNN) (Salinas *et al.,* 2020). The fundamental difference between classical forecasting techniques such as ARIMA and ETS with DeepAR is that classical techniques apply one model to each individual time series. On the other hand, DeepAR has the ability to combine several similar time series to create a model that can be used for many time series cases. Study conducted by Salinas *et al.,* (2020) stated that DeepAR outperforms the ARIMA and exponential smoothing (ETS) methods. Additionally, DeepAR can generate forecasts for time series similar to those it was trained on, although this particular study only utilized one time series and no covariates.

The value of time series i at time t is denoted by $z_{i,t}$. With given past data

$$[z_{i,1}, \dots, z_{i,t_{o-2}}, z_{i,t_{o-1}}] := \mathbf{z}_{i,1:t_{o-1}} \tag{8}$$

the conditional distribution

$$P(\mathbf{z}_{i,t_o:T} | \mathbf{z}_{i,t_{o-1}}, \mathbf{x}_{i,1:T}) \tag{9}$$

of the future of each time series

$$[z_{i,t_0}, z_{i,t_{0+1}} \dots, z_{i,T}] := \mathbf{z}_{i,t_o:T} \tag{10}$$

here t_0 denotes the time point from which we assume that $z_{i,t}$ is unknown at prediction time. We assume that the model distribution

$$Q_\Theta(\boldsymbol{z}_{i,t_o:T} | \boldsymbol{z}_{i,1:t_{0-1}}, \boldsymbol{x}_{i,1:T}) \tag{11}$$

consist of a product of likelihood factors

$$\prod_{t=t_o}^{T} Q_\Theta(\boldsymbol{z}_{i,t_o:T} | \boldsymbol{z}_{i,1:t-1}, \boldsymbol{x}_{i,1:T}) = \prod_{t=t_o}^{T} p(z_{i,t} | \theta(\mathbf{h}_{i,t}, \Theta)) \tag{12}$$

parameterized by the output $\mathbf{h}_{i,t}$ of an autoregressive recurrent network

$$\mathbf{h}_{i,t} = h(\mathbf{h}_{i,t-1}, z_{i,t-1}, x_{i,t}, \Theta) \tag{13}$$

where h is a function implemented by a multi-layer recurrent neural network with LSTM cells parametrized by Θ. The network output $\mathbf{h}_{i,t}$ into a function $\theta(\mathbf{h}_{i,t}, \Theta)$, which then builds the parameters of the fixed distribution $p(z_{i,t} | \theta(\mathbf{h}_{i,t}))$.

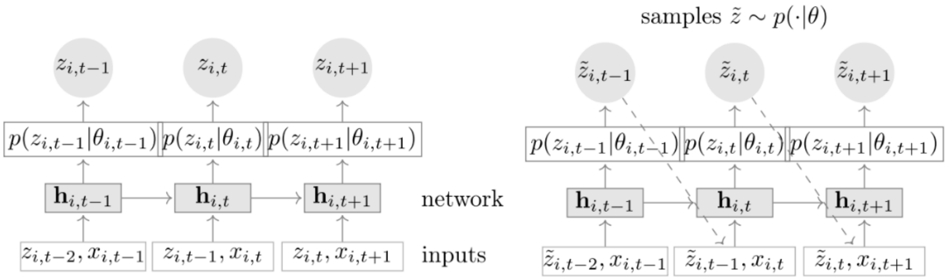

Figure 1: Summary of DeepAR Model (Salinas *et al.,* 2020).

The model is autoregressive, which means that it uses the observation at the last step $z_{i,t_{o-1}}$ as an input, and is also recurrent, i.e., the previous output is fed back as an input $h_{i,t_{o-1}}$ at the next step.

2.4.2. Model Training

Salinas *et al.,* (2020) elaborated that during the training, the parameter Θ of the model consists of the parameters of the RNN $h(\cdot)$ and the parameter of $\theta(\cdot)$. They can be learned by maximizing the log-likelihood.

$$\mathcal{L} = \sum_{i=1}^{N} \sum_{t=t_0}^{T} \log p(z_{i,t} | \theta(\mathbf{h}_{i,t})) \tag{14}$$

Since $\boldsymbol{h}_{i,t}$ is a deterministic function of the input, all quantities needed to compute (14) are observed, so it can be optimized directly via stochastic gradient descent.

Salinas *et al.,* (2020) also emphasized that a network with parameters Θ has three inputs: the covariates $x_{i,t}$, the target value of the previous time step $z_{i,t_{o-1}}$, and the network output of the previous time step $\mathbf{h}_{i,t-1}$. The network output

$$\mathbf{h}_{\mathrm{i,t}} = h(\mathbf{h}_{\mathrm{i,t-1}}, z_{i,\mathrm{t-1}}, \mathbf{x}_{i,t}, \Theta) \tag{15}$$

is then used to calculate the parameters

$$\theta_{\mathrm{i,t}} = \theta(\mathbf{h}_{\mathrm{i,t}}, \Theta) \tag{16}$$

of the likelihood $(z|\theta)$, which is used to train the model parameters. For prediction, the history of the time series $z_{i,\mathrm{t}}$ is fed into the network for $t < t_o$, then a sample is drawn in the prediction domain for $t \geq t_o$ and fed back for the next point until the end of the prediction range $t = t_o + T$. Repeating this prediction process results in multiple traces that represent the joint predicted distribution.

2.4.3. Hyperparameter in DeepAR

Hyperparameters are parameters whose values control the learning process and determine the values of model parameters that a learning algorithm ends up learning. There are several hyperparameter in DeepAR, such as layers, cells, cell type, dropout rate, epoch, batch size, workers, learning rate. Five parameters were chosen for this research.

Learning rate is a parameter in an optimization algorithm that controls the adjustment of weights with respect to the loss gradient (Khanam and Foo, 2021). A learning rate that is too low will result in slower convergence to the minimum loss, while a learning rate that is too high can cause the algorithm to overshoot and miss the optimal solution.

The number of layers (Num layers) refers to the count of hidden layers in a neural network. Data enters through the input layer, passes through the hidden layers for processing, and finally exits through the output layer. The data must propagate through the specified number of layers (Choudhary and Kesswani, 2020). The number of workers (Num workers denotes the number of parallel processes employed during training. Increasing the number of workers can enhance parallelization and improve training efficiency (Giuseppe *et al.,* 2019).

An epoch represents how many times a dataset is used for training. Specifically, one epoch means the entire dataset is passed through the model once. Different datasets may require varying numbers of epochs for optimal training (Siami-Namini *et al.,* 2018). The batch size is the number of data samples used in each iteration during an epoch to train the network. Setting this hyperparameter too high can lengthen the training process and slow convergence, while setting it too low can lead to oscillations without achieving satisfactory performance (Kendel and Castelli, 2020).

2.4.4. Integrating Optuna Hyperparameter Optimization with DeepAR

Optuna uses the define-by-run principle, allowing users to dynamically construct the search space within the optimization framework. Unlike the define-and-run principle, which restricts the manipulation of intermediate variables once the network is defined, define-by-run does not require separate phases for network construction and computation. This flexibility enables more adaptive and efficient optimization.

The cost-effectiveness of a hyperparameter optimization framework hinges on two key factors: the efficiency of parameter selection, determining which parameters to investigate, and the performance estimation strategy, which evaluates the value of currently investigated parameters based on learning curves and determines parameters to discard (Akiba *et al.,* 2019).

Optuna incorporates both types of sampling methods: relational sampling, which utilizes correlations among parameters, and independent sampling, which handles each parameter separately. Independent sampling methods like tree-structured Parzen estimator (TPE) and relational sampling methods such as covariance matrix adaptation evolution strategy (CMA-ES) are integrated within Optuna.

For performance estimation, Optuna employs a pruning algorithm that operates in two phases. Firstly, it periodically monitors intermediate objective values and terminates trials that fail to meet predefined conditions. Secondly, Optuna utilizes a variant of the asynchronous successive halving algorithm (ASHA), which enables aggressive early stopping based on the provisional ranking of trials. The combination of efficient searching and pruning algorithms solidifies Optuna as the primary option to address hyperparameter tuning within the context of the DeepAR method.

2.5 Evaluation Metrics

2.5.1. Root Mean Square Error (RMSE)

Root mean squared error is a common statistical metric to measure model performance in many academic studies and to measure how well the forecasted values compared to the observed values. RMSE penalizes variance as it gives errors with larger absolute values more weight than errors with smaller absolute values. (Chai and Draxler 2014). It is defined as the square root of the mean of all squared errors, or the standard deviation of forecasting error. The formula of RMSE as stated by Chai and Draxler (2014) is

$$\mathrm{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N} e_i^2} \tag{17}$$

where e denotes the differences between actual and forecasted value and n denotes the amount of observation data. Lower RMSE values indicate better forecasting accuracy.

2.5.2. Mean Absolute Percentage Error (MAPE)

Mean absolute percentage error is the average absolute error between the forecasted value and the actual value. It is commonly used metric to evaluate the accuracy of forecasting or prediction model, due to its advantages of scale-independency and interpretability (Kim and Kim, 2016). The formula of MAPE as stated by Kim and Kim (2016) is

$$\text{MAPE} = \frac{1}{N}\sum_{t=1}^{N}\left|\frac{A_t - F_t}{A_t}\right| 100\% \tag{18}$$

where A_t and F_t denote the actual and forecast values at data point t and N is the number of data points. The best model is the one that has the smallest MAPE value.

2.6 Data Analysis Procedure

The stages of the analysis carried out in the study are as follows:

1. Data Exploration: Initial exploration of rice production data is performed by visualizing it through line chart representations.
2. Data Partitioning: The data is divided into training data spanning 2018-2021 and test data for the year 2022.
3. SARIMA Modelling:
   a) Seasonal Determination: The seasonality parameter (s) is established based on the seasonal patterns present in the data.
   b) Stationarity Testing: Stationarity tests for the mean and variance are conducted utilizing the Augmented Dickey Fuller (ADF) test and the Box-Cox test, respectively.
   c) Data Transformation: If the data does not adhere to the stationarity assumption, appropriate transformations are applied to achieve stationarity.
   d) Order Identification: Optimal p and q orders for the SARIMA model are determined through analysis of the Autocorrelation Function (ACF).
   e) Extended Identification: Further identification is performed using the extended sample ACF (EACF) plot, aiming to select potential orders (p, q). This plot may sometimes yield two potential orders.
   f) Model Selection: The optimal SARIMA model is selected based on the Akaike information criterion (AIC) and the corrected Akaike information criterion (AIC_C) ensuring the best balance between model complexity and goodness-of-fit.
4. DeepAR Modelling:
   a) The dataset is formatted into JSON, incorporating monthly seasonal patterns to facilitate the DeepAR model's analysis.
   b) Hyperparameter Tuning: The Optuna library, an automatic hyperparameter tuning framework, is utilized to optimize model parameters. This includes tuning the learning rate from 0.001 to 0.1, adjusting the number of layers from 1 to 5, varying the number of workers from 2 to 5, setting batch sizes between 4 and 48, and aiming to minimize both root mean square error (RMSE) and mean absolute percentage error (MAPE).
   c) Model Selection: The superior model is selected based on MAPE and RMSE evaluation metrics.
5. Intrepreting the result: MAPE and RMSE metrics between SARIMA and DeepAR models were compared. The performance of the DeepAR model is compared against the SARIMA model using the MAPE and RMSE metrics. This comparison provides a clear evaluation of the model’s effectiveness in predicting the given dataset.

6. Result Interpretation: The findings from the performance comparison were interpreted to draw meaningful conclusions and insights. The results were analyzed to understand the strengths and limitations of each model.

3. Result and Discussion

3.1 Data Exploration

The data utilized in this study pertain to monthly rice production, extracted from the BPS Book titled "Luas Panen dan Produksi Beras" spanning the years 2018 to 2022, encompassing the period from January 2018 to December 2022, thus comprising 60 observations.

Due to the customary and climatic considerations, farmers commence rice field cultivation at the beginning of the rainfall season, occurring between October and March. Consequently, March emerges as the month with the most rice production in Indonesia.

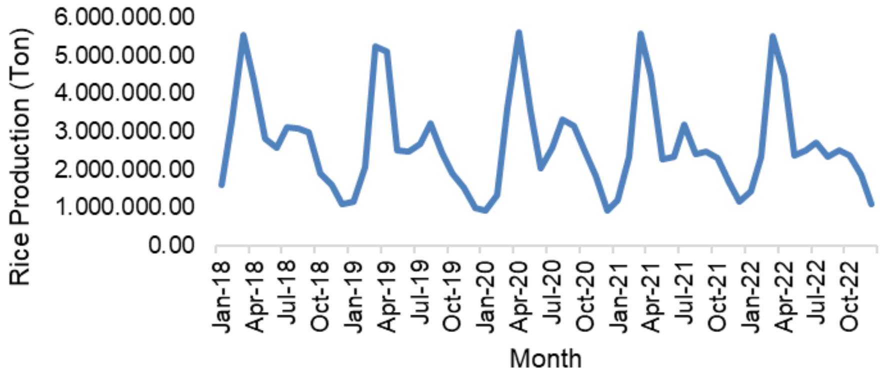

Figure 2: Monthly Rice Production in Indonesia 2018-2022 (Ton).

The data shows annual cycle, culminating in March with a production peaked at approximately 5,000,000 tonnes and experiencing a dip in December and January, where it hovers around 1,000,000 tonnes. This recurring pattern supports the designation of March-April as "Panen Raya," a term signifying a bountiful harvest celebration.

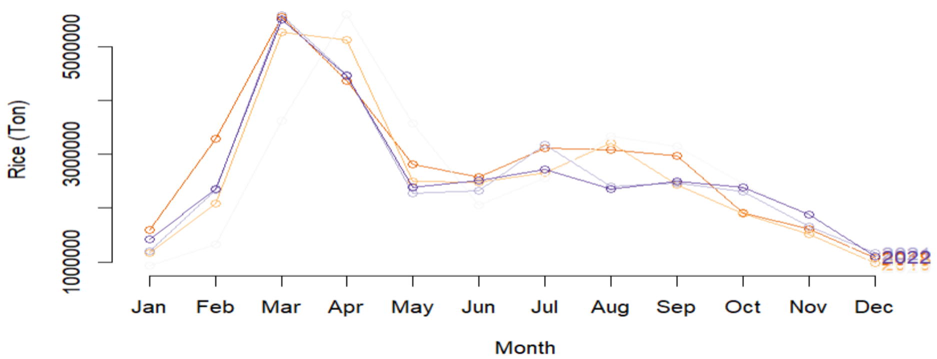

Figure 3: Monthly Rice Production in Indonesia by Year.

Given the discerned annual cycle within the data, it becomes clear to use forecasting methods capable of accommodating seasonal patterns to predict the data trend. Consequently, the DeepAR and SARIMA models have been selected for comparison in Indonesia’s rice production data. The dataset will be partitioned into four years for training and one year for testing. The division of time periods for analysis is flexible, as long as it aligns with the natural yearly cycle of rice cultivation. Notably, the absence of missing values or outlier data makes it unnecessary for data imputation in this research.

3.2 DeepAR

DeepAR, available in the GluonTS library, can be executed using either Torch or MXNet. However, the gluon-ts.torch module lacks a set_seed() function, resulting in varying outputs with each execution. Consequently, using GluonTS with MXNet is preferable for achieving consistent results. To stabilize the outputs, set.seed(7) was applied, although this seed can be adjusted as needed to meet specific requirements. The parameters used for DeepAR include num_layers, num_workers, batch_size, learning_rate, and epochs.

Table 1: Top 10 DeepAR's Hyperparameter with Lowest MAPE.

No	Num Layer	Num worker	Learning Rate	Batch Size	MAPE	RMSE	Epoch
1	**4**	**4**	**0,002792057**	**28**	**4,50%**	**148820,6**	**50**
2	3	3	0,029528025	34	4,82%	160537,2	50
3	1	3	0,022186597	13	4,99%	192501,4	50
4	2	5	0,016816644	4	5,22%	223456,3	25
5	4	4	0,022424286	6	5,34%	170569,9	50
6	1	5	0,009838659	4	5,41%	160258,3	50
7	2	4	0,010793208	22	5,79%	187500,6	25
8	2	4	0,046643941	13	6,06%	199495,9	25
9	4	2	0,006743020	47	6,12%	185457,4	50
10	5	4	0,021991919	5	6,14%	181860,6	25

Hyperparameter tuning methods like grid search, random search, and bayesian optimization offer different approaches to finding optimal hyperparameters for machine learning models. Optuna, the library employed here, stands out by employing Bayesian optimization algorithms, including tree-structured Parzen estimator (TPE) and Gaussian process-based optimization. Optuna streamlines the hyperparameter tuning process by automating the exploration of the hyperparameter space, ultimately enabling researchers and practitioners to efficiently find the most effective configurations for their models.

3.3 SARIMA

In determining the autoregressive (AR) and moving average (MA) parameters in seasonal autoregressive integrated moving average (SARIMA) models using R, there are two primary methods. SARIMA models are crucial for analyzing and forecasting seasonal data as they account for both non-seasonal and seasonal components, making them highly relevant for time series data exhibiting regular patterns.

The first method involves several preparatory steps, beginning with the assessment of stationarity through the augmented Dickey-Fuller (ADF) test and the Box-Cox test. Stationarity is a critical assumption in ARIMA and SARIMA models because non-stationary data can lead to unreliable results (Montgomery *et al.,* 2015). If the data do not exhibit stationarity, a root transformation is applied to achieve stationarity. In this instance, transforming the data with the power of 0.6 resulted in the data meeting the stationarity assumption.

Once the data pass the stationarity test, the next step is to determine the non-seasonal AR, Integration (I), and MA parameters. This is achieved using the autocorrelation function (ACF) and extended autocorrelation function (EACF) in R.

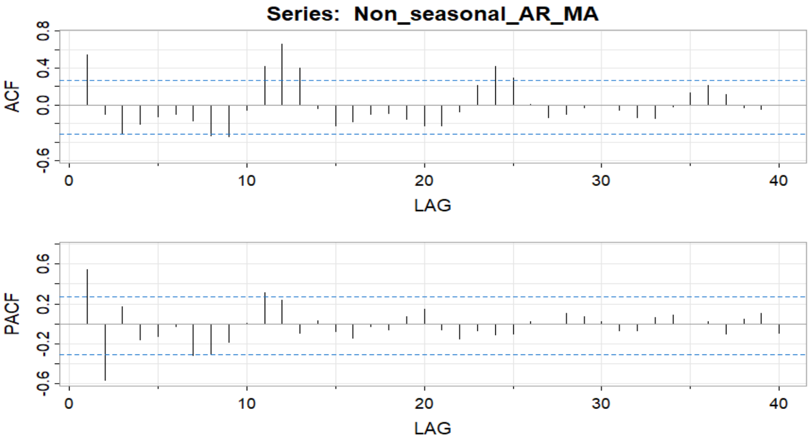

Figure 4: Plot ACF and PACF of Non-Seasonal AR and MA.

AR/MA	0	1	2	3	4	5	6	7	8	9	10	11	12	13
0	x	o	x	o	o	o	o	x	x	o	x	x	x	o
1	x	o	x	o	o	o	o	o	x	o	o	x	x	o
2	o	o	o	o	o	o	o	o	o	o	o	x	o	o
3	x	o	o	o	o	o	o	o	o	o	o	x	o	o
4	x	o	o	o	o	o	o	o	o	o	o	o	o	o
5	o	o	x	o	o	o	o	o	o	o	o	o	o	o
6	o	x	x	o	o	o	o	o	o	o	o	o	o	o
7	x	o	o	o	o	o	o	o	o	o	o	o	o	o

Figure 5: EACF Plot of Non-Seasonal AR and MA.

Based on Figure 4 and Figure 5, the ACF, PACF, and EACF revealed that the tentative models ARIMA(0,0,1), ARIMA(2,0,0), ARIMA(2,0,1), and ARIMA(0,0,3) were selected for further analysis. Figure 4 also shows a pattern of non-stationarity at lags 12, 24, 36, and 48, indicating a seasonal component. Therefore, seasonal differencing was needed to address the non-stationarity issue.

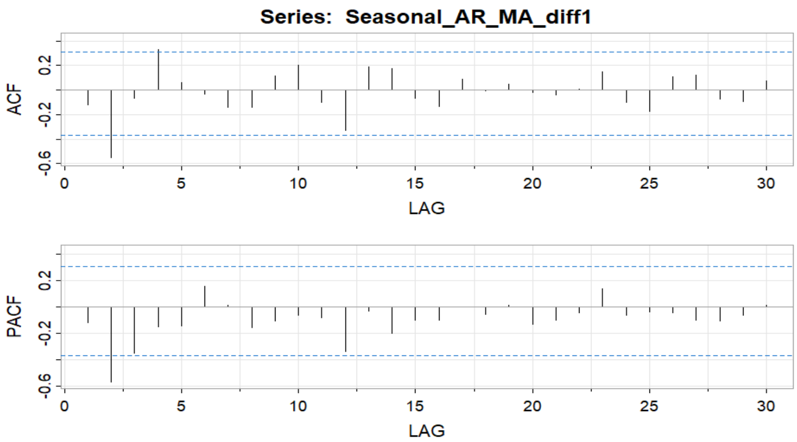

Figure 6: ACF and PACF Plot of Seasonal AR and MA.

Utilizing the ACF and PACF, the model selected for the seasonal component was ARIMA(0,1,0)$_{12}$. The next step involves evaluating the best model using the AIC and the corrected Akaike information criterion (AIC_C). Through this evaluation, the two best models identified were ARIMA(0,0,1)(0,1,1)$_{12}$ and ARIMA(2,0,1)(0,1,0)$_{12}$.

The second method employs the auto.arima function from the "forecast" library in R to automatically determine the best parameters for ARIMA and SARIMA models. The automatic method offers the advantage of reducing the complexity and subjectivity involved in manual model selection. The result of this function indicated that the optimal model was ARIMA(0,0,1)(0,1,1)$_{12}$. After identifying potential model using ACF, PACF, EACF, and auto.arima() function, overfitting were used to find the best model.

Table 2: Overfitting Model Candidates for SARIMA.

Model	**AIC**	**AIC$_C$**
ARIMA(2,0,0)(0,1,1)$_{12}$	1061,35	1062,64
ARIMA(2,0,1)(0,1,1)$_{12}$	1063,12	1065,12
ARIMA(0,0,3)(0,1,1)$_{12}$	1063,62	1065,62

Utilizing AIC and AIC_C, the model selected for model comparison and selection were **ARIMA(2,0,0)(0,1,1)$_{12}$, ARIMA(2,0,1)(0,1,1)$_{12}$, ARIMA(0,0,3)(0,1,1)$_{12}$.**

3.4 Models Comparison and Selection

In this research, two metrics used to evaluate model performance are root mean square error (RMSE) and mean absolute percentage error (MAPE). MAPE measures the accuracy of the forecast as a percentage, providing an easy-to-understand error rate, while RMSE gives an absolute measure of the forecast error magnitude, which is useful for understanding the variability in the errors. Both metrics were obtained by comparing the forecasted data from the selected models to the actual data.

Table 3: ARIMA Models with The Lowest MAPE.

Model	**MAPE**	**RMSE**
ARIMA(0,0,3)(0,1,1)$_{12}$	**5,44%**	**346894,7**
ARIMA(0,0,1)(1,1,0)$_{12}$	5,88%	346939,5

ARIMA(2,0,0)(0,1,1)$_{12}$	6,09%	351995,7

Table 4: DeepAR Models with The Lowest MAPE.

Num Layer	Num worker	Learning Rate	Batch Size	MAPE	RMSE	epoch
4	**4**	**0,002792057**	**28**	**4,50%**	**148820,6**	**50**
3	3	0,029528025	34	4,82%	160537,2	50
1	3	0,022186597	13	4,99%	192501,4	50

All models exhibit acceptable MAPE values, being below 10%. However, the DeepAR models demonstrate superior performance in forecasting rice production in Indonesia, outperforming the SARIMA (ARIMA(0,0,3)(0,1,0)$_{12}$) models in both RMSE and MAPE metrics. This suggests that the DeepAR model, with its ability to capture complex temporal dependencies, is more effective for this type of forecasting task.

The superior performance of DeepAR could be attributed to its deep learning architecture, which allows it to model intricate patterns and seasonal effects more effectively than traditional statistical models like SARIMA. This is particularly important in the context of rice production in Indonesia, where factors such as "panen raya" (harvest season) introduce significant variability into the data.

Previous studies have also highlighted the challenges in forecasting agricultural production due to seasonal variations, weather conditions, and other external factors (Divisekara *et al.,* 2020) using SARIMA. Incorporating these insights, our study confirms that advanced neural network models like DeepAR can provide more accurate and reliable forecasts compared to traditional methods.

Below are graphical comparison between the actual data, the forecasted data from the SARIMA models, and the forecasted data from the DeepAR models, illustrating the superior accuracy of the DeepAR forecasts.

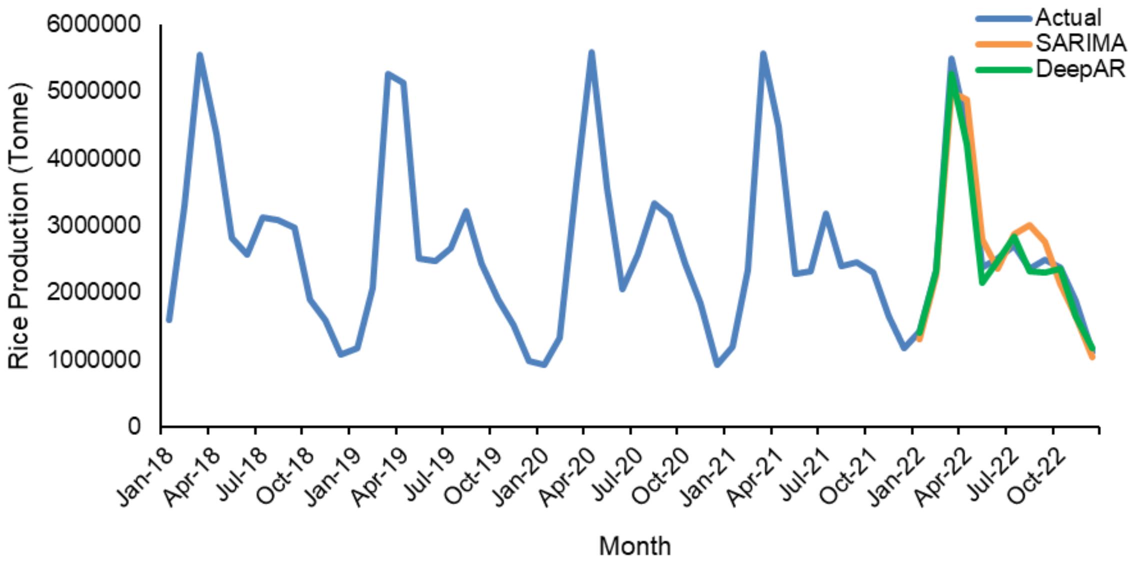

Figure 7: Comparison of Actual Data with Forecast of SARIMA and Forecast of DeepAR.

Upon closer examination, The DeepAR Forecast demonstrated superior predictive performance for the April and August better than SARIMA. This enhanced accuracy during these critical months contributes to the overall superior performance, both in RMSE and MAPE, of the best DeepAR model over the best SARIMA model.

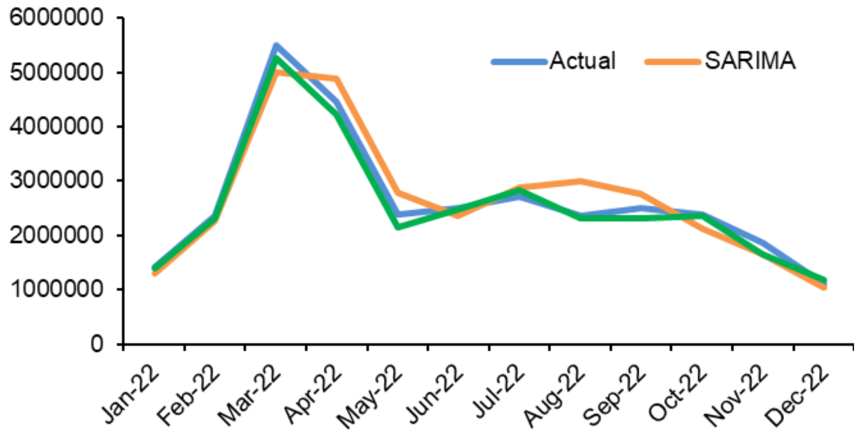

Figure 8: Comparison of Data Test with Forecast of SARIMA and Forecast of DeepAR.

March and April are peak months of production due to the planting season starting at the beginning of the rainy season, as shown in Figure 8. During these months, the optimal growing conditions provided by the abundant rainfall lead to higher yields. The practical implications of these findings are significant for stakeholders in the agriculture sector. Accurate forecasts can help policymakers plan and manage rice supply effectively, ensuring food security during these high-yield periods. Additionally, supply chain managers can optimize logistics, reduce waste, and plan storage more efficiently to handle the increased production volumes. Future research could further refine these models by incorporating additional variables and exploring other machine learning approaches to continue improving forecasting accuracy.

4. Conclusion and Advice

The monthly rice production consistently exhibits an annual cyclic pattern attributable to its susceptibility to climatic variations. Notably, March and April are acknowledged as the "Panen Raya" months within this cyclic pattern. Due to the observed seasonality, the rice production dataset is suitable for analysis using seasonal autoregressive integrated moving average (SARIMA) and DeepAR models. Employing mean absolute percentage error (MAPE) and root mean squared error (RMSE) as evaluation metrics, it is deduced that DeepAR (num_layer = 4, num_worker = 4, learning_rate = 0,002792057, batch_size = 28, epoch = 50) surpasses ARIMA$(0,0,3)(0,1,1)_{12}$ in the context of rice production forecasting. Specifically, DeepAR achieves a MAPE of 4.5% and an RMSE of 148820,62 whereas SARIMA yields a MAPE of 5,44% and an RMSE of 346894,47.

The DeepAR model used in this study is implemented through the GluonTS package. However, it is noteworthy that Amazon has introduced an enhanced method of this model known as DeepAR+ within the platform. It is regrettable that access to requires financial assistment, as it is a subscription-based service. For prospective research endeavors, it is advisable to consider employing DeepAR+ if the financial resources are available. The advanced features and optimizations incorporated into DeepAR+ within the environment could potentially yield superior results.

Furthermore, it is necessary to highlight a technical constraint pertaining to the Mxnet package, which is essential for the execution of DeepAR. Regrettably, the current version of Mxnet does not offer support for GPU acceleration, thereby limiting the potential for faster computational runtime. Mxnet also have not been updated since Python 3.9.x. This limitation should be taken into consideration when contemplating the computational efficiency of the DeepAR model in the context of this study.

Another important consideration is the necessity to specify the number of data points to be forecasted from the outset when using MXNet, as it is not possible to save the model state. This limitation is compounded by the use of the Optuna framework for hyperparameter optimization, in which you cant even save the exact learning rate values. Therefore, it is crucial to define the forecast horizon at the beginning of the modeling process, as adjustments cannot be made post analysis.

Reference

Aceto, G., Ciuonzo, D., Montieri, A., Persico, V., & Pescapé, A. (2019). Know your Big Data trade-offs when classifying encrypted mobile traffic with deep learning. 2019 Network Traffic Measurement and Analysis Conference (TMA), 121–128. https://doi.org/10.23919/TMA.2019.8784565

Akiba, T., Sano, S., Yanase, T., Ohta, T., & Koyama, M. (2019). Optuna: A next-generation hyperparameter optimization framework. Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2623–2631. https://doi.org/10.1145/3292500.3330701

Bang, S., Bishnoi, R., Chauhan, A. S., Dixit, A. K., & Chawla, I. (2019). Fuzzy logic-based crop yield prediction using temperature and rainfall parameters predicted through ARMA, SARIMA, and ARMAX models. 2019 Twelfth International Conference on Contemporary Computing (IC3), 1–6. https://doi.org/10.1109/IC3.2019.8844871

Be, E. (2022). Impact of rice production, rice supply, and rice prices on food security in North Timor District. Jurnal Manajemen & Agribisnis, 19(3), 407. https://journal.ipb.ac.id/index.php/jma/article/view/12345

Choudhary, S., & Kesswani, N. (2020). Analysis of KDD-Cup'99, NSL-KDD and UNSW-NB15 datasets using deep learning in IoT. Procedia Computer Science, 170, 1561–1573. https://doi.org/10.1016/j.procs.2020.03.367

Cryer, J. D., & Chan, K. S. (2008). Time series analysis with applications in R (2nd ed.). New York, NY: Springer.

Divisekara, R. W., Jayasinghe, G. J. M. S. R., & Kumari, K. W. S. N. (2020). Forecasting the red lentils commodity market price using SARIMA models. SN Business & Economics, 1(20). https://doi.org/10.1007/s43546-020-00020-x

Fattah, J., Ezzine, L., Aman, Z., El Moussami, H., & Lachhab, A. (2018). Forecasting of demand using ARIMA model. International Journal of Engineering Business Management, 10, 1–9. https://doi.org/10.1177/1847979018808673

Forsberg, H. M. (2018). ‘If they do not want to work and suffer, they must starve and die.’ Irish and Finnish famine historiography compared. Scandinavian Journal of History, 43(4), 484–514. https://doi.org/10.1080/03468755.2018.1467077

Khanam, J. J., & Foo, S. Y. (2021). A comparison of machine learning algorithms for diabetes prediction. ICT Express, 7(4), 432–439. https://doi.org/10.1016/j.icte.2021.02.004

Li, Y., Cao, Y., Yang, J., Wu, M., Yang, A., & Li, J. (2024). Optuna-DFNN: An Optuna framework driven deep fuzzy neural network for predicting sintering performance in big data. Alexandria Engineering Journal, 97, 100–113. https://doi.org/10.1016/j.aej.2023.11.013

Moghar, A., & Hamiche, M. (2020). Stock market prediction using LSTM recurrent neural network. Procedia Computer Science, 170, 1168–1173. https://doi.org/10.1016/j.procs.2020.03.049

Montgomery, D. C., Jennings, C. L., & Kulahci, M. (2015). Introduction to time series analysis and forecasting (2nd ed.). New Jersey, NJ: John Wiley & Sons.

Ohyver, M., & Pudjihastuti, H. (2018). ARIMA model for forecasting the price of medium quality rice to anticipate price fluctuations. Procedia Computer Science, 135, 707–711. https://doi.org/10.1016/j.procs.2018.08.235

Patria, D. G., & Sukamto, S. (2021). Rice science and technology. Malang: Literasi Nusantara.

Rahim, I., Zulfikar, Z., & Kafrawi, K. (2017). Teknik budidaya dan tingkat produksi tanaman padi sawah peserta dan non-peserta sekolah lapang pengendalian hama terpadu di Kabupaten Pinrang. Agroplantae: Jurnal Ilmiah Terapan Budidaya dan Pengelolaan Tanaman Pertanian dan Perkebunan, 6(2), 1–12. https://doi.org/10.51978/agro.v6i2.36

Salinas, D., Flunkert, V., Gasthaus, J., & Januschowski, T. (2020). DeepAR: Probabilistic forecasting with autoregressive recurrent networks. International Journal of Forecasting, 36(3), 1181–1191. https://doi.org/10.1016/j.ijforecast.2019.07.001

Shumway, R. H., & Stoffer, D. S. (2016). Time series analysis and its applications with R examples. Springer Texts in Statistics.

Siami-Namini, S., Tavakoli, N., & Namin, A. S. (2018). A comparison of ARIMA and LSTM in forecasting time series. 2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA), 1394–1401. https://doi.org/10.1109/ICMLA.2018.00227

Sugiura, N. (1978). Further analysis of the data by Akaike’s information criterion and the finite corrections. Communications in Statistics-Theory and Methods, 7(1), 13–26. https://doi.org/10.1080/03610927808827599

Sumarsono. (2015). Ketahanan pangan beras: Faktor penyebab & kontribusi wilayah. LPPM UNHASY Tebuireng Jombang.

Yoo, T. W., & Oh, I. S. (2020). Time series forecasting of agricultural products' sales volumes based on seasonal long short-term memory. Applied Sciences, 10(22), 8169. https://doi.org/10.3390/app10228169