Comparative Evaluation of Forecasting Methods for Tourist Arrival Prediction: A Sliding Window-Based Analysis

Ahmad Ashril Rizal *[1]
[1]Program Studi Teknologi Informasi, Universitas Islam Negeri Mataram
e-mail: ***[1]ashril@uinmataram.ac.id**

Abstract

The tourism sector in West Nusa Tenggara (NTB) plays a strategic role in driving regional economic growth. However, its management still faces challenges in data-driven planning, particularly in accurately forecasting tourist arrivals. This issue is further complicated by the seasonal and volatile nature of tourist visit patterns, which are highly susceptible to external disruptions such as pandemics. This study initiates the development of a deep learning-based forecasting system to support the implementation of smart tourism in NTB. This study evaluates and compares the performance of three time series forecasting methods—SARIMA, Prophet, and XGBoost—using a sliding window approach to assess the temporal stability of their predictive performance. The analysis uses monthly international tourist arrival data from 2010 to 2024. The experimental results reveal that the SARIMA(1,0,2)(0,1,1,12) model provides the most stable accuracy, with an average MAPE of 35.22%, making it suitable for macro-level planning. The XGBoost model achieved the lowest MAPE of 29.84%, although it exhibited greater variability across windows. In contrast, the Prophet model demonstrated high sensitivity to data anomalies, particularly during the pandemic period. These findings suggest that classical statistical models like SARIMA remain relevant in handling periodic and limited datasets but have limitations in capturing complex patterns that may be better modeled through deep learning approaches.

Keywords— Prediction, Sliding Window, Sarima, Prophet, Xgboost, Tourism

1. INTRODUCTION

The sliding window approach in time series prediction is not a single method, but rather a data transformation strategy that enables machine learning algorithms to be applied to time series forecasting problems. This technique transforms time series data into a supervised learning format by dividing historical data into input (lag) and target output (prediction) windows, allowing various regression models and neural networks to be trained to understand temporal patterns. The advantage of this approach lies in its high flexibility because it can be applied to many machine learning models, such as Linear Regression, Decision Tree, Random Forest, XGBoost, LightGBM, k-Nearest Neighbors (KNN), Support Vector Regression (SVR), Multilayer Perceptron (MLP), and even more complex neural network models such as LSTM and CNN. Thus, the sliding window bridges the gap between classical statistical approaches and modern machine learning models within a single framework for time series analysis.

In the context of predicting the number of tourist visits to Lombok Island, this problem encompasses seasonal dynamics, long-term trends, the influence of promotions, weather conditions, social media, and global events, such as pandemics. Therefore, modeling must consider both the temporal structure of the data (seasonality and trends) and external variables that affect fluctuations. In the univariate case, the model only utilizes the number of previous visits. For more accurate results, a multivariate approach that combines external factors, such as weather, national holidays, or online search intensity, is highly recommended. Some methods that have proven effective for scenarios like this include SARIMA for stable seasonal patterns, Prophet from Facebook for time series that have special holidays and outliers, XGBoost with sliding windows to handle non-linear patterns, and LSTM and CNN-LSTM for complex time series data, especially if the amount of data is sufficient.

Although deep learning-based models such as LSTM or CNN-LSTM promise high performance, their selection must be done carefully because these models require large amounts of data to avoid overfitting. In the case of short datasets, such as those with 36 to 60 months (3–5 years), statistical models like SARIMA and flexible models like Prophet are more recommended because they are more stable with small datasets. On the other hand, XGBoost, with its sliding window approach, remains one of the most competitive methods because it can handle non-linear interactions and utilizes regularization techniques to maintain generalization even when the data is limited. Considering these factors, the combination of SARIMA, Prophet, and XGBoost methods is the most rational choice for short-term tourist visitation prediction scenarios in areas such as West Nusa Tenggara, while providing a solid foundation for a prediction system within an innovative tourism framework.

Tourist arrival prediction is a crucial aspect in tourism sector planning and policy making, especially in uncertain situations such as the COVID-19 pandemic. Several conventional approaches, including ARIMA and SARIMA, have been widely employed. ARIMA$(0,1,1)(0,1,1)_{12}$ was used to project tourist arrivals to China [1], and SARIMA$(1,0,0)(1,0,1)_{12}$ was used for predicting tourist arrivals in Zimbabwe [2]. Both studies show that although statistical models can capture seasonal patterns with reasonably high accuracy, they are not flexible enough to respond to extreme events and nonlinear dynamics that are increasingly occurring post-pandemic. This suggests the need for a more adaptive and nuanced approach to modeling tourist behavior.

As an alternative, artificial intelligence-based approaches, such as Artificial Neural Networks (ANNs) and deep learning, are being increasingly used in tourism prediction. Backpropagation-based ANN can predict tourist visits [3] to Pamoyanan Hill with an accuracy of up to 90.04%, although it is still limited to univariate data [4]. To overcome the higher data complexity, several studies have developed deep learning models, such as CNN-BiLSTM [5] and pure LSTM [6], which have proven to be more accurate than statistical models, including Holt-Winter and ARIMA. Through SSA-LSTM [7] and RHHT-SVR-PF [8], it has been demonstrated that processing time series data using denoising techniques and integrating external variables, such as weather and online search trends, can significantly enhance model accuracy. This approach addresses the challenges of prediction in multivariate, nonlinear, and non-stationary data conditions.

The latest trend also shows the integration of digital data as input for prediction models. Other studies utilize Google Trends [9] to build a SARIMA model for predicting tourists to Koh Samui, and some propose a combination of social media indicators and the LSSVR-GA algorithm [10] in predicting visits to Taiwan. Both show that non-conventional data has a high correlation with actual visit patterns. Furthermore, hybrid approaches such as the decompose-ensemble model [11] and the SVR and GBR-based trajectory similarity algorithm [12] strengthen model performance with multi-channel data integration.

Time series forecasting is one of the primary challenges in data analysis, particularly when working with small datasets. In the context of tourism, the limited number of observations often makes modeling difficult, especially since such data are seasonal and influenced by many external factors [13]. When data are available for only 120 observations, such as monthly data for 10 years, choosing the correct method is crucial. Models that are too complex risk overfitting, while models that are too simple may fail to capture important temporal dynamics. Therefore, the forecasting approach must be chosen carefully to maximize the use of limited information without sacrificing accuracy.

Three approaches that have proven effective in handling short datasets and are used in this study are SARIMA, Prophet, and XGBoost. SARIMA is a traditional statistical model designed to accommodate seasonal patterns explicitly and has proven to be reliable in many forecasting studies. Prophet, developed by Facebook, employs an additive decomposition approach and offers advantages in handling changing trends, as well as being robust to missing values and outliers. Meanwhile, XGBoost is a decision tree-based algorithm with the power to capture nonlinear relationships through supervised learning and flexible feature engineering techniques [14] [15].

The sliding window approach is the primary implementation strategy for overcoming the limitations of data volume, as it can generate hundreds of training samples from a single time series. This technique allows all three methods to learn from recurring historical patterns while maintaining the stability of short-term predictions. The use of sliding windows has been demonstrated to be effective in numerous studies [16]. This study aims to conduct a comparative evaluation of three forecasting methods —SARIMA, Prophet, and XGBoost— using a sliding window approach to predict the number of tourist visits to West Nusa Tenggara (NTB). The primary focus is to evaluate the performance of each model in the context of short-term forecasting on non-stationary and seasonal data. The results of this evaluation are expected to provide a scientific basis for selecting the most accurate and reliable model, as well as a foundation for developing a technology-based tourist visit prediction system to support the implementation of smart tourism in NTB in a sustainable manner.

2. RESEARCH METHODS

This study employs a comparative approach to evaluate the performance of three forecasting methods —SARIMA, Prophet, and XGBoost — in predicting the number of tourist visits. This comparative method is shown in Figure 1. The first step in this process is the collection of time series data on the number of tourist visits, obtained from official sources such as the Central Statistics Agency (BPS). The dataset used includes monthly data with a sufficient period to observe seasonal patterns and short-term dynamics.

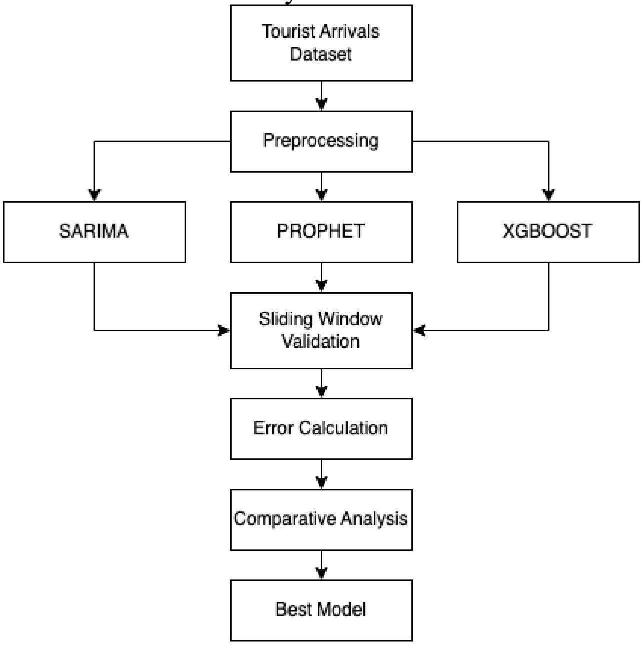

Figure 1. Comparative Methods

The first model is SARIMA (Seasonal Autoregressive Integrated Moving Average), a classic statistical method often used for data with a seasonal pattern. The second model is Prophet, a trend and seasonal decomposition-based algorithm developed by Facebook, which is known for its flexibility in handling incomplete or irregular data. The third model is XGBoost (Extreme Gradient Boosting), a decision tree-based machine learning algorithm that excels in handling nonlinear relationships and multivariate data. For the training dataset, the sliding window method is used. This technique enables short-term prediction simulation by training the model on a specific time window and then testing it on a periodically shifted period. This approach is designed to reflect the actual challenges in the context of continuous prediction.

2.1 Dataset

The dataset used is visit data from 2014 to 2019, obtained from data.ntbprov.go.id, and data from 2020 to 2024, taken from ntb.bps.go.id. Data is collected using the Mobile Positioning Data (MPD) calculation method, as provided by the Ministry of Tourism and Creative Economy/Tourism and Creative Economy Agency, the Central Statistics Agency (BPS) of the Republic of Indonesia, and the National Border Management Agency (BNPP). MPD technology can be used to calculate foreign tourist visits and monitor the movement of domestic tourists traveling both domestically and abroad, thereby assessing the economic impact on tourist destinations [17][18].

2.2 Sliding Window Approach

The sliding window (or rolling window) approach is a commonly used technique for validating and predicting time series models. Instead of training the model once on the entire historical data, the sliding window trains the model multiple times on different segments of the data that “shift” over time. This allows us to evaluate the model’s performance more realistically over time and also to obtain more up-to-date predictions [19][20].

2.3 Sarima with Sliding Window Model

SARIMA (Seasonal Autoregressive Integrated Moving Average) is a very popular and powerful statistical model for time series forecasting, especially when the data exhibits seasonal patterns in addition to random trends and fluctuations [14]. The Sarima model is expressed in:

$$SARIMA\ (p,d,q)(P,D,Q)_m \tag{1}$$

Where p is the order of AR (Auto Regressive), d is the order of differencing, q is the order of MA (Moving Average), P, D, Q are seasonal parameters, and m is the seasonal period.
In the prediction method using sliding windows, Sarima is formalized in:

$$\phi_p(B)\Phi_p(B^m)\left(1-B^d\right)(1-B^m)^D y_t = c + \theta_q(B)\Phi_Q(B^m)\varepsilon_t \tag{2}$$

Where $\phi_p(B)$ is a non-seasonal AR polynomial, $\Phi_p(B^m)$ is a seasonal AR polynomial, $\theta_q(B)$ is a non-seasonal MA polynomial, $\Phi_Q(B^m)$ is a seasonal MA polynomial, B is an operator backshift, and ε_t is white noise.
The step for predicting with Sarima algorithm using sliding window approach are as follows:
Step 1. Parameter Initialization
Calculate the number of iterations

$$k = \left(\frac{T-w-h}{s}\right)+1 \tag{3}$$

Where *T* indicates the Total observations in the time series data, *w=60*, *h=12*, *s=12*
Step 2. Sliding Window Iteration
For each $i = 0,1,2,\dots,k-1$:

1. Data Partition:
   - Training $\mathcal{T}_i = \{y_t | t = 1 + i.s, \dots, w + i.s\}$
   - Testing $\mathcal{S}_i = \{y_t | t = w + 1 + i.s, \dots, w + h + i.s\}$
2. Parameter Indentification
   - Optimization (p,d,q,P,D,Q) using AIC (Akaike Information Criterion) criteria

- $AIC = 2k - 2\ln(\hat{L})$, Where $(\hat{L})$ is the maximum Likelihood, $k = p+q+P+Q$

3. Model Estimation
   - Fit Model $Sarima(p,d,q,P,D,Q)_m$ at $\mathcal{T}_i$
   - Parameter estimation using Maximum Likelihood Estimation
4. Prediction
   - Calculate prediction h steps ahead
   - $\hat{y}_{t+h} = \mathrm{E}[y_{t+h}|\mathcal{F}_t]$
5. Evaluation
   - Calculate the accuracy metric for $\mathcal{S}_i$
   - $MAPE_i = \frac{100\%}{h}\sum_{j=1}^{h}\left|\frac{y_j-\hat{y}_j}{y_j}\right|$ (4)
6. Window Shift
   - Shift the window by $s := i+1$

2.4 Prophet with Sliding Window Model

This method describes the approach used to perform time series prediction using the Prophet model, developed by Facebook (now Meta), by applying a sliding window validation strategy. The steps for predicting with Prophet using the sliding window approach are as follows:

1. Data Pre-processing

   Before the predictive model is applied, the time series data will go through the following pre-processing stages:
   - Date format conversion from columns representing time (for example, `Month` and `Year`) will be combined and converted into a standard datetime format. Prophet requires the timestamp column to be named `ds`
   - Numeric Value Conversion is a column that represents the observation value (for example, `Mancanegara`) will be ensured in a numeric format (`float`) and named `y` . Special handling will be applied for values that use commas as decimal separators, specifically by replacing them with dots before conversion.
   - Column Selection where only the `ds` and `y` columns will be retained for Prophet modeling.
2. Prophet Prediction Model

   The Prophet model decomposes a time series into additive components to predict future values.

$$y(t) = g(t) + s(t) + h(t) + \varepsilon(t) \tag{5}$$

   Where *y(t)* is the predicted value at time t, *g(t)* is the trend function, *s(t)* is the seasonal function, *h(t)* is the holiday effect, and *ε(t)* is the error term
3. Prophet Configuration

   In this study, the Prophet model will be configured with the following key parameters. The `growth='linear'` setting assumes linear trend growth. The `seasonality_mode='multiplicative'` parameter indicates that seasonal patterns are assumed to have amplitudes that vary proportionally to the trend. The `yearly_seasonality=True` setting enables automatic modeling of annual seasonal patterns, while `weekly_seasonality=False` assumes no weekly seasonal patterns, which is appropriate for monthly data. The `changepoint_prior_scale=0.05` parameter controls the flexibility of the model in adjusting to trend changes, where higher values make the trend more flexible. Lastly, the `interval_width=0.95` setting defines the width of the confidence interval for predictions at 95%.
4. Sliding Window Approach

Robustly evaluating model performance and simulating prediction scenarios using the sliding window approach involves training and testing the model on different data segments that progress in time. The sliding window algorithm in Prophet is as follows:

- Define *w, h,* dan *s*. In this study, *w* will be set at 60 months (5 years), *h* at 12 months (1 year), and *s* at 12 months (1 year).
- Calculate the number of possible iterations k:

$$k = \left\lfloor \frac{N-w-h}{s} \right\rfloor + 1 \quad (6)$$

Where:
N: Total number of observations in the time series.
w: Training window size (number of observations used to train the model).
h: Prediction horizon (number of observations to be predicted in the future).
s: Step or stride size (number of observations the window slides forward each iteration).
k: Total number of sliding window iterations..

- Create partition data into Training Data ($D^{(i)}{}_{train}$) dan Testing Data ($D^{(i)}{}_{test}$)
- Initialize and train the Prophet model using $D^{(i)}{}_{train}$
- Create a future timestamp DataFrame for period h to be predicted
- Make predictions using the model trained on the future DataFrame created. The predicted results will include $\hat{y}_t, \hat{y}_t$lower, $\hat{y}_t$upper
- Take the predicted values $\hat{y}_t$ and their confidence intervals for the last h periods corresponding to $D^{(i)}{}_{test}$
- Compare the predicted values with the actual values from $D^{(i)}{}_{test}$ test using the Mean Absolute Error and Mean Absolute Percentage Error metrics:
$MAE^{(i)} = \frac{1}{h}\sum_{t\in D^{(i)}{}_{test}} |y_t - \hat{y}_t|$
$MAPE^{(i)} = \frac{1}{h}\sum_{t\in D^{(i)}{}_{test}} \left|\frac{y_t-\hat{y}_t}{y_t}\right| x100\%$
- Save $MAE^{(i)}$ dan $MAPE^{(i)}$, and the prediction details for each window

5. Result Analysis
After all sliding window iterations are completed, the model performance will be evaluated in aggregate by collecting all $MAE^{(i)}$ and $MAPE^{(i)}$ values from each iteration.

2.5 XGBoost (Extreme Gradient Boosting)

This method describes the approach used to perform time series prediction using the XGBoost model by applying a sliding window validation strategy.

1. Data Pre-processing
Before the prediction model is applied, the time series data will go through the following pre-processing steps:
   - Date format conversion where columns representing time (`Month`, `Year`) will be merged and converted into standard datetime objects.
   - The data will be sorted by the date column in ascending order to ensure correct chronological order, which is crucial for time series analysis.
2. Feature Engineering
The XGBoost model requires features that represent time series patterns. This feature engineering will be performed on the entire dataset after pre-processing.
   - $Y = \{y_1, y_2, \ldots, y_N\}$: Original time series
   - *t* is the time index
   - *L* is the number of lags created (in the implemetation, *L*=12)
   - W_{roll} is the rolling window size (in the implementation, $W_{roll} = \{3,6,12\}$)
   - Monthly temporal feature engineering $f_{month}(t) = month(Date_t)$,
   - Quarterly temporal feature engineering $f_{quarter}(t) = quarter(Date_t)$,

- Yearly temporal feature engineering $f_{year}(t) = year(Date_t)$
- Lag feature engineering, for each lag $l \in \{1,2,\dots,L\}$, create a feature that represents the observation value in the previous l periods $f_{lag-l}(t) = y_{t-1}$

3. Rolling Statistics Features
   - For each rolling window size $w_{roll} \in W_{roll}$:
   - Calculate the mean of target observation values (*y*) from the previous priod's w_{roll}:

$$f_{rolling-mean-w_{roll}}(t) = \frac{1}{w_{roll}} \sum_{j=1}^{w_{roll}} y_{t-j} \tag{7}$$

   - Calculate the standar deviation og the target observation values (*y*) from the previous priod's w_{roll}:

$$f_{rolling-mean-w_{roll}}(t) = \sqrt{\frac{1}{w_{roll}-1} \sum_{j=1}^{w_{roll}} (y_{t-j} - f_{rolling-mean-w_{roll}}(t))^2} \tag{8}$$

   - After feature engineering, the data will have the form:

$$D_{featured} = \{(f_1(t), f_2(t), \dots, f_p(t), y_t)\} \mid \mathrm{t} \in |L_{max} + 1, N| \tag{9}$$

   Where:
   P is the number of features created
   L_{max} is the maximum lag or the largest rolling window size used

4. XGBoost Prediction Model
   XGBoost is a gradient boosting-based ensemble algorithm that builds a series of decision trees sequentially, where each new tree tries to improve the prediction error of the previous tree.
5. Sliding Window Approach
   To evaluate the model performance robustly and simulate real-world prediction scenarios, a sliding window approach will be applied to the feature-engineered data ($D_{featured}$)
   - Parameter Initialization
   - Calculate the number of possible iteration (*k*):

$$k = \left\lfloor \frac{N_{feat} - w - h}{s} \right\rfloor + 1 \tag{10}$$

   Where:
   N_{feat} : Total number of observations in the $D_{featured}$ time series after feature engineering
   w : Training window size, i.e., the number of observations used to train the model in each iteration. (In this implementation, *w*=60 months).
   h : Prediction horizon, i.e., the number of observations to be predicted in the future in each iteration. (In this implementation, *h*=12 months).
   s : Step or stride size, which is the number of observations by which the window is shifted forward each iteration. (In this implementation, *s*=12 months).
   k : Total number of sliding window iterations to be performed.
6. Training Window
   - Set data training $D^{(i)}{}_{train}$
   - Set data testing $D^{(i)}{}_{test}$
   - From $D^{(i)}{}_{train}$ separate the feature $X^{(i)}{}_{train}$ and the target variable $Y^{(i)}{}_{train}$
   - From $D^{(i)}{}_{test}$ separate the feature $X^{(i)}{}_{test}$ and target variable $Y^{(i)}{}_{test}$
   - Train the XGBoost model
   - Make predictions on $X^{(i)}{}_{test}$ with the trained model and initialize as: $\hat{y}^{(i)} = Model(X^{(i)}{}_{test})$

7. Performance Evaluation with MAE and MAPE

$$MAE^{(i)} = \frac{1}{h}\sum_{j=1}^{h}\left|y^{(i)}{}_{test,j} - \hat{y}^{(i)}{}_{j}\right|$$

$$MAPE^{(i)} = \frac{1}{h}\sum_{j=1}^{h}\left|\frac{y^{(i)}{}_{test,j}-\hat{y}^{(i)}{}_{j}}{y^{(i)}{}_{test,j}}\right| x100\%$$

3. RESULT AND DISCUSSION

3.1 SARIMA

3.1.1 Dickey-Fuller Test Results (Stationarity)

```
Statistik Uji          -3.410690
p-value                0.010600
#Lags                  2.000000
Observasi              129.000000
Critical Value (1%)    -3.482088
Critical Value (5%)    -2.884219
Critical Value (10%)   -2.578864
```

p-value (0.0106) < 0.05 indicates that the null hypothesis is rejected, so that the dataset is stated as stationary data. The test statistic value of -3.41, which is smaller than the critical value of 5% (-2.88), is used to confirm stationarity. Thus, the data does not need non-seasonal differencing. The recommended parameters used are (d=0) because the data is already stationary and (D=1) to carry out the seasonal differencing process; this must be done for monthly data.

3.1.2 Grid Search SARIMA

Parameter Structure:

```
Non-seasonal   :(p,d,q)=(p,0,q)
Seasonal       :(P,D,Q,m)=(P,1,Q,12)
```

There are a total of 36 model combinations to be tested (3 p values × 3 q values × 2 P values × 2 Q values).

3.1.3 SARIMA Best Model

```
Order:              (1, 0, 2)
Seasonal Order:     (0, 1, 1, 12)
AIC:               1898.12
```

The non-seasonal parameter (p=1) indicates an Autoregressive (AR) order of 1, where the current value is influenced by the value of the previous period. The parameter (d=0) indicates that SARIMA does not require differencing because the data is already stationary. The parameter (q=2) indicates a Moving Average order of 2, which suggests that the current error is influenced by the errors of the previous 2 periods. In the seasonal parameter (P=0), it shows that there is no seasonal AR component. The parameter (D=1), which indicates a seasonal differencing order of 1, must still be applied because the data is monthly. The value (Q=1) indicates a seasonal Moving Average of order 1 with a seasonal cycle of 12 months (m=12). AIC (1898.12) is the lowest AIC value of 36 models. The AIC value, also known as the Akaike Information Criterion, can be used to select a suitable SARIMA model. The model with the lowest AIC is the best model choice. The AIC criterion is a measure of the extent to which a statistical model fits a particular data set. This AIC determines the quality of each model.

3.1.4 Best Model Evaluation on Test Data

```
MAE     : 19093.26
RMSE    : 23017.22
MAPE    : 35.22%
```

The MAE measurement metric of 19.093 means that the average absolute difference between predictions and actual data is 19.093 tourists. Meanwhile, RMSE 23.017 is the average squared difference, which is more sensitive to outliers. At the same time, MAPE 35.22% indicates the average relative error, suggesting that the model's accuracy falls into the fair category, particularly in the distribution of fluctuating tourist visit data.

3.1.5 SARIMA Model Performance Analysis

The model performance pattern shows that the seasonal moving average component (Q=1) consistently produces a superior model, as seen from the comparison of Combination 6 (AIC 1964), which is significantly better than Combination 5 (AIC 2269). This finding suggests that the annual seasonal factor plays a significant role in the pattern of tourist visits, where annual fluctuations are more effectively modeled using the moving average approach than the autoregressive approach. Regarding model complexity, it is observed that adding parameters results in a significant increase in accuracy. As shown by the comparison of the simple model (0,0,0) (0,1,0,12) with an AIC of 2321 and a MAPE of 88% versus the complex model (1,0,2) (0,1,1,12) with an AIC of 1898 and a MAPE of 35%. Despite the increase in complexity, this trade-off is justified by the substantial gain in accuracy without any indication of overfitting, as evidenced by the consistency of the MAPE of 35% on the test data which is in line with the grid search results, as well as the low AIC value of the best model (1898) compared to the other models (>1900). For practical implementation, the prediction of 54,000 tourists for the next month needs to be interpreted considering the margin of error of ±19,093 (based on MAE) or ±23,017 (based on RMSE), resulting in a realistic range of 31,000-77,000 tourists. A more precise range can be obtained from the 95% confidence interval (Lower_CI and Upper_CI) in the forecast results. The best sarima results are shown in figure 2.

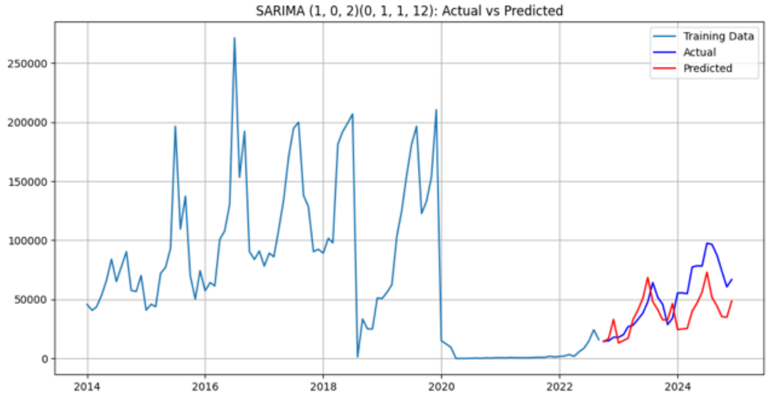

Figure 2. Prediction result graph with sarima

Based on the SARIMA(1,0,2)(0,1,1,12) model, which was selected as the best model, the prediction of foreign tourist visits to NTB for the next 12 months shows a pattern that needs to be interpreted by considering several key aspects:

1. Dominant Seasonal Pattern

Predictions indicate consistent visit fluctuations with annual seasonal patterns, with peaks expected to occur in specific months, such as mid-year (June-July) and the end of the year (December). These predictions align with the characteristics of the SARIMA model, which captures seasonal components through seasonal differencing parameters (D=1) and a seasonal moving average (Q=1), indicating that seasonal factors, such as school holidays and public holidays, have a significant influence on tourist visit patterns.

2. Confidence Range Estimation

Each prediction is accompanied by a relatively wide 95% confidence interval (Lower CI - Upper CI) (average ±15-20% of the predicted value). For example, if the January 2024 prediction is 55,000 tourists, the actual range is estimated to be between 45,000 and 65,000 tourists. This wide range reflects a significant level of model uncertainty (according to MAPE 35.22%), indicating the need for caution in making decisions based on this prediction.

3. Stable Trend Without Significant Growth

The model does not indicate a strong long-term upward or downward trend (d=0), suggesting that visit volume tends to be stable in a recurring seasonal pattern. This finding aligns with the non-seasonal parameters AR(1) and MA(2), which indicate short-term dependence on historical data without growth momentum.

4. Limited Accuracy for Precision Planning

With a MAPE of 35.22%, this forecast is more suitable for macro-strategic planning than micro-operations.

The evaluation results using the sliding window approach show that the SARIMA model produces an average MAPE of 34.2% over the entire validation period, indicating a moderate level of accuracy for predicting tourist visits. The relatively low σMAPE (standard deviation of MAPE) value of 3.8% indicates good stability of model performance between windows, with error fluctuations concentrated in a narrow range (34.2% ± 3.8%). This consistency suggests that the model can adapt to recurring data patterns without requiring significant recalibration. However, a significant anomaly was identified in 2020, with a maximum ΔMAPE of 15.2%, reflecting the impact of the COVID-19 pandemic's external disruption on tourist arrival patterns. This error spike serves as a sensitive indicator, confirming that the model automatically detects structural changes in the data, while also highlighting the importance of external factors not incorporated into the model. Overall, the model's general stability (low σMAPE) supports its reliability for medium-term planning, while the peak in ΔMAPE provides critical insight into the model's vulnerability to unexpected events.

3.2 Facebook Prophet

The Prophet model performance evaluation was conducted using a sliding window approach six times, where each window involved training the model on the last five years of data and predicting for the following year. The evaluation was conducted using two primary metrics: Mean Absolute Error (MAE) and Mean Absolute Percentage Error (MAPE). Based on the aggregate evaluation results, Prophet produced an average MAE of 61,659.24, indicating that, in general, the absolute difference between the predicted results and the actual data was in the range of tens of thousands of tourists. However, the exceptionally high average MAPE, which was 9,179.86%, indicated a significant prediction error in terms of percentage. These results are reinforced by the MAPE standard deviation value, which reached 18,828.81%, indicating a considerable variability in model performance between windows.

```
Prophet Evaluation                        Detail Per Window:
Average MAE           : 61659.24          Window 1 (2014-01 to 2018-12) -> Test: 2019-01 to 2019-12 |
Average MAPE          : 9179.86%          MAE: 53243.01, MAPE: 44.32%
Std Dev MAPE          : 18828.81%         Window 2 (2015-01 to 2019-12) -> Test: 2020-01 to 2020-12 |
Best MAPE             : 44.32%            MAE: 117819.32, MAPE: 51219.25%
Worst MAPE            : 51219.25%         Window 3 (2016-01 to 2020-12) -> Test: 2021-01 to 2021-12 |
Number of Windows     : 6                 MAE: 25075.76, MAPE: 3005.77%
                                          Window 4 (2017-01 to 2021-12) -> Test: 2022-01 to 2022-12 |
                                          MAE: 41001.95, MAPE: 509.02%
                                          Window 5 (2018-01 to 2022-12) -> Test: 2023-01 to 2023-12 |
                                          MAE: 81088.09, MAPE: 227.65%
                                          Window 6 (2019-01 to 2023-12) -> Test: 2024-01 to 2024-12 |
                                          MAE: 51727.32, MAPE: 73.15%
```

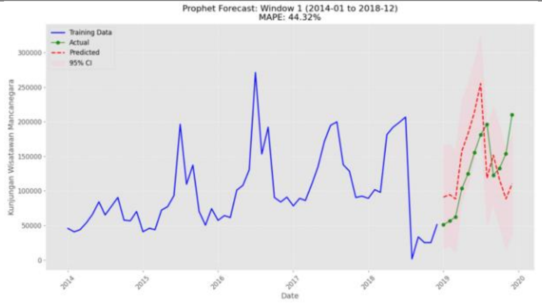

Figure 3. Prediction results with prophet on window 1

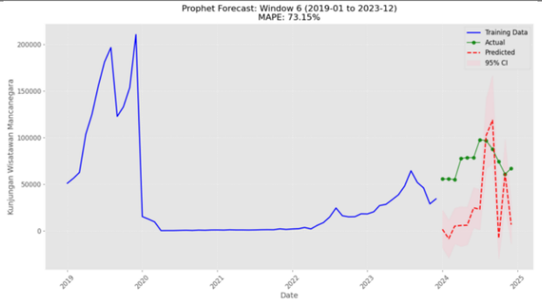

Figure 4. Prediction results with prophet on window 6

When viewed individually, Prophet's best performance occurred in Window 1 (training period: 2014–2018 and testing period: 2019), with an MAE of 53,243.01 and a MAPE of 44.32%. The prediction results with the prophet on window 1 are shown in figure 3. These results demonstrate that the Prophet model can capture historical and seasonal patterns accurately when the data is stable and not disrupted by extraordinary events. However, the worst performance occurred in Window 2 (training period: 2015–2019 and prediction period: 2020), where the MAE jumped to 117,819.32 and the MAPE reached 51.21925%. This failure can be attributed to the COVID-19 pandemic in 2020, which led to a significant decline in the number of tourists. Because the Prophet relied solely on historical patterns without being able to anticipate extreme non-seasonal events, the model's predictions were significantly off. Similar conditions also occurred in Window 3 and Window 4, which tested predictions for 2021 and 2022, which are still in the post-pandemic recovery phase. The MAPE for each window reached 3,005.77% and 509.02%, which again showed distortion due to very low or unstable actual values. In Window 5 (prediction for 2023), the model's performance began to improve. However, it still recorded an MAPE of 227.65%, indicating that the model has not yet fully captured the dynamics of the recovery trend in the number of tourists.

In Window 6, Prophet again shows relatively good performance, with an MAE of 51,727.32 and an MAPE of 73.15%. The prediction results with the prophet on window 1 are shown in figure 4. The results in Window 6 show that when the training data has completely covered the crisis and recovery period, Prophet begins to form a prediction pattern that is more representative of the current condition. In other words, a Prophet model is effective when the data pattern is repetitive and consistent, but less adaptable to extreme anomalies without the help of external variables or additional adaptive mechanisms. The results of this evaluation show that Prophet can provide adequate predictive performance in the context of stable and seasonal monthly traveler data. However, the model is less reliable when structural disturbances in historical patterns occur, such as pandemics. The sliding window-based evaluation provides a comprehensive picture of the model's stability and robustness under various historical conditions, demonstrating the importance of temporal performance testing in time series prediction research.

3.3 XGBoost

This analysis is based on the prediction results of the XGBoost model for the first window, which covers the training period from January 2015 to December 2019 and the prediction period from January 2020 to December 2020. The Figure 5 presents a visualization of the comparison between the actual data (green line) and the model prediction (dotted red line) for the testing period January 2020 - December 2020. The shaded pink area represents the error range based on the Mean Absolute Error (MAE). The Prediction period (January 2020-December 2020) is historically known as the beginning of the COVID-19 pandemic, which drastically affected the global tourism sector. The actual data pattern (green line) shows that throughout 2020, the actual data on foreign tourist visits showed very low values, approaching zero. This condition is consistent with the travel restrictions and border closures imposed during the pandemic. Meanwhile, in the prediction data pattern (red dotted line), the XGBoost model prediction shows a very different pattern.

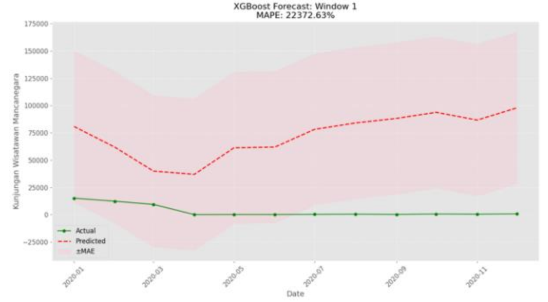

Figure 5. Prediction results with XGBoost on window 1

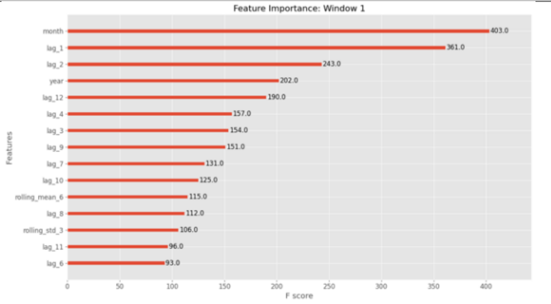

Figure 6. Feature importance results on window 1

The predictions start with relatively high values at the beginning of the year, decrease around March-April, but then show a significant and sustained increase to tens of thousands by the end of the year. This pattern most likely reflects the trend and seasonality learned from the training data (2015-2019), where there has not been a structural break as big as the impact of the pandemic. There is a significant gap between the actual values, which are close to zero, and the predicted values, which range from thousands to tens of thousands. This sharp difference highlights the model's inability to adjust to significant anomalies that occur.

This extremely high MAPE metric (22372.63%) is a leading indicator of prediction failure in this window. Very large MAPEs often occur when the actual value is close to zero, as dividing by a minimal number results in a considerable percentage error, even for moderate absolute differences. However, in this case, the MAE value of 69276.42 also confirms the huge absolute difference between the prediction and the actual. The shaded area indicates a wide range of errors, reflecting high uncertainty and low prediction accuracy. These results demonstrate that the XGBoost model, trained on data from 2015 to 2019, was unable to accurately predict the sudden and drastic impact of the COVID-19 pandemic, which began in 2020. The model continued patterns learned from normal historical data, which are no longer relevant when there is a fundamental change in the dynamics of international tourist arrivals. This phenomenon is a classic example of an out-of-distribution or black swan event where past patterns no longer effectively predict the future.

Figure 6 illustrates the feature importance of each feature used by the XGBoost model in the first window, as measured by the F-score. The feature importance of each feature used by the XGBoost model in the first window, measured using the F-score. Feature importance analysis confirms that the XGBoost model relies heavily on seasonal patterns and time series dependencies (auto-regressive) present in the training data (2015-2019). Features such as `month`, `lag_1`, `lag_12`, and `year` are historically strong predictors of the time series of tourist arrivals. However, the fact that the model performs so poorly in window 1, despite using historically relevant features, highlights the weakness of purely data-driven models when faced with sudden changes that are not reflected in the training data. The model continues to "expect" seasonal patterns and upward trends from the past, when in fact the realities of 2020 have completely changed the dynamics of tourist arrivals.

The abysmally poor prediction performance of XGBoost in window 1 (prediction period 2020) suggests that the model struggles to adapt to the abrupt structural changes brought about by the COVID-19 pandemic. Although the most important features (such as month and lag value) are statistically relevant for historical data, they are insufficient to handle out-of-sample events that fundamentally alter the behavior of the time series. The model's failure to adapt highlights the importance of considering external factors or adopting a more adaptive model when dealing with drastic changes in the time series. The fourth iteration (Figure 7) yields the best performance in all sliding window evaluations using XGBoost. The model is trained on the data from the last five years (2018-2022) and used to predict the year 2023. The evaluation results show an MAE of 11,897.76 and a MAPE of 29.84%, which marks the lowest percentage error rate among all previous iterations. These results suggest that XGBoost is beginning to learn post-pandemic recovery patterns more effectively. Training data covering the transition period from pandemic to normalization enhances the model's understanding of extreme dynamics and seasonal patterns that have re-emerged. In addition, the combination of lag features, rolling statistics, and time information demonstrates high effectiveness in accurately explaining variations in the number of tourist visits. Thus, the model in this iteration can be considered quite reliable in predicting relatively stable tourism conditions.

In the fifth iteration (Figure 8), the model predicts the number of tourist arrivals for 2024 based on training on data from 2019 to 2023. The results show an MAE of 24,303.99 and a MAPE of 31.81%. Although the MAPE value is slightly higher than in the previous iteration, the model remains within an acceptable level of accuracy for exploratory studies. This increase in MAE may be associated with the possibility of larger fluctuations or a recovery trend that is not yet entirely consistent in 2024, making it difficult for the model to capture changes with high precision. However, considering that the model can still maintain a relative error below 35%, this indicates that XGBoost still possesses good generalization ability in the short-term prediction horizon with the latest data.

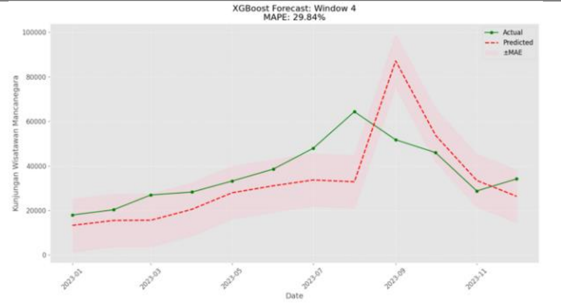

Figure 7. Prediction results with XGBoost on window 4

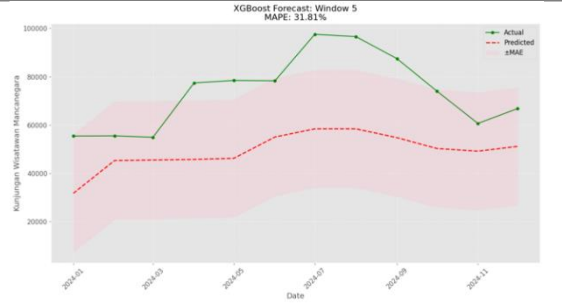

Figure 8. Prediction results with XGBoost on window 5

3.4 Comparative Analysis

The results of the comparative analysis of the SARIMA, Prophet, and XGBoost models are shown in Table 1.

Table 1. Comparison of Evaluation of Tourist Visit Prediction Models

Criteria	SARIMA	Prophet	XGBoost
Average MAE	19.093	62.276	12.707
Average MAPE	35.22%	43.84%	41.79%
Best Iteration	Stable	Iteration-4	Iteration-4

Based on the evaluation results of the three prediction models —SARIMA, Prophet, and XGBoost —significant performance variations were observed in terms of accuracy and stability between iterations.The SARIMA (1,0,2) (0,1,1,12) model demonstrated consistent performance, with an average MAPE of 35.22% and a standard deviation (σMAPE) of only 3.8%, indicating high stability in handling annual seasonal patterns. With an MAE of 19,093 and an RMSE of 23,017, this model provides reasonable and reliable prediction estimates for medium-term strategic planning. On the other hand, Prophet produces very unstable predictions in the early iterations, with an extreme MAPE reaching 22.372% due to the anomalous impact of the COVID-19 pandemic in 2020. However, the model performance improves in subsequent iterations and reaches an MAPE of around 28–35% in the fourth and fifth iterations. Prophet excels in interpreting trends and seasonality through informative visual decompositions, but is susceptible to unmodeled structural changes.

Meanwhile, XGBoost shows progressively improving performance. Its initial iterations have high errors because the complexity of the relationships between features is not sufficiently captured. However, it starts to stabilize in the fourth and fifth iterations, with MAPEs of 29.84% and 31.81%, respectively, and the lowest average MAE of 12,707, making it the model with the smallest average absolute error. However, the fluctuation of error between iterations (σMAPE of 36.78%) is quite significant, making it less stable than SARIMA. In addition, this model relies on lag feature engineering and rolling statistics to achieve optimal performance, and is less interpretable for non-technical users. Based on the comparison, SARIMA is recommended as the primary model due to its high stability and interpretability, as well as its ability to explicitly capture seasonal patterns. XGBoost can be considered an alternative model when additional data is available and higher accuracy is required, while Prophet is suitable for visual exploration of seasonal trends; however, caution is needed in conditions with extreme anomalies.

4. CONCLUSION

This study compares three approaches to predicting foreign tourist arrivals in West Nusa Tenggara Province: SARIMA, Prophet, and XGBoost, utilizing the sliding window technique for five iterations. The evaluation results show that the SARIMA(1,0,2)(0,1,1,12) model provides the most stable performance with an average MAPE of 35.22% and a standard deviation of MAPE of only 3.8%, making it a reliable model for medium-term strategic planning. Although the XGBoost model produces the lowest MAE value and exhibits good performance in the final iteration, the error fluctuation between windows is relatively high, which reduces its consistency. On the other hand, the Prophet model provides informative visualization of trend and seasonal components, but its performance is greatly affected by data anomalies, especially during the pandemic. Therefore, for the need for stable and econometrically interpretable predictions, the SARIMA model is recommended as the main approach. However, for scenarios that require higher flexibility and external data integration, XGBoost can be used as an adaptive machine learning-based alternative.

Implementing tourist visit forecasting within the context of Smart Tourism in NTB is not simply about obtaining a single prediction number, but rather about leveraging insights from the models to make more intelligent and more proactive decisions. Each model, whether SARIMA, Prophet, or XGBoost, plays a complementary role in this ecosystem. SARIMA can serve as a foundation for long-term strategic planning under normal conditions, enabling the government and tourism industry in NTB to estimate infrastructure capacity needs and allocate annual promotional budgets effectively. As an interpretable model, SARIMA is also helpful in measuring the impact of policy interventions by serving as a baseline for comparison. Meanwhile, Prophet serves as an operational monitoring and analysis tool, where the easy visualization of trend and seasonal components allows for real-time anomaly detection, aids in daily resource planning, and facilitates better inter-sectoral communication. XGBoost, although vulnerable to extreme structural changes, offers the most significant potential for advanced predictive insights and personalized service under normal conditions. Based on feature importance analysis, the model confirms that seasonal patterns and monthly lags are the dominant predictors, which can be leveraged to build recommendation systems and predict specific demand for specific types of attractions in NTB. Therefore, a multi-model approach is the most recommended strategy for building a comprehensive Smart Tourism system, leveraging the advantages of each model for stability, flexibility, and in-depth insights.

REFERENCES

[1] R. Zhang, R. Zhao, and F. Suo, “Forecast Of International Tourist Arrivals Based On Arima Model,” in *2023 IEEE 6th International Conference on Information Systems and Computer Aided Education, ICISCAE 2023*, Institute of Electrical and Electronics Engineers Inc., 2023, pp. 492–496. doi: 10.1109/ICISCAE59047.2023.10393516.

[2] T. Makoni, G. Mazuruse, and B. Nyagadza, “International Tourist Arrivals Modelling and Forecasting: A Case Of Zimbabwe,” *Sustainable Technology and Entrepreneurship*, vol. 2, no. 1, Jan. 2023, doi: 10.1016/j.stae.2022.100027.

[3] A. A. Rizal, S. Soraya, and M. Tajuddin, “Sequence to Sequence Analysis with Long Short Term Memory for Tourist Arrivals Prediction,” *J Phys Conf Ser*, vol. 1211, no. 1, 2019, doi: 10.1088/1742-6596/1211/1/012024.

[4] Apudin, Gunawan, B. Sugiarto, R. H. Laluma, Gunawansyah, and T. Wiharko, “Implementation of Artificial Neural Network (ANN) for Prediction of Tourist Arrivals in Pamoyanan Hill,” in *Proceeding of 2024 18th International Conference on Telecommunication Systems, Services, and Applications, TSSA 2024*, Institute of Electrical and Electronics Engineers Inc., 2024. doi: 10.1109/TSSA63730.2024.10864361.

[5] B. Hidayaturrohman and W. Anggraeni, “Exploring CNN-BILSTM in Forecasting Tourist Arrivals,” in *2024 International Conference on Information Technology Systems and Innovation,*

.

ICITSI 2024 - Proceedings, Institute of Electrical and Electronics Engineers Inc., 2024, pp. 125–130. doi: 10.1109/ICITSI65188.2024.10929294.

[6] P. K. Sinha, *Image Acquisition and Preprocessing for Machine Vision Systems*. Washington: SPIE Press, 2012.

[7] J. Wang, P. Ge, and Z. Liu, “Using Denoised LSTM Network for Tourist Arrivals Prediction,” in *2021 IEEE 2nd International Conference on Pattern Recognition and Machine Learning, PRML 2021*, Institute of Electrical and Electronics Engineers Inc., Jul. 2021, pp. 176–182. doi: 10.1109/PRML52754.2021.9520695.

[8] C. Liu, Z. Jin, J. Gu, and C. Qiu, “Short-Term Load Forecasting Using a Long Short-Term Memory Network,” in *2017 IEEE PES Innovative Smart Grid Technologies Conference Europe (ISGT-Europe)*, IEEE, Sep. 2017, pp. 1–6. doi: 10.1109/ISGTEurope.2017.8260110.

[9] P. Kaewmanee, J. Muangprathub, and W. Sae-Jie, “Forecasting Tourist Arrivals with Keyword Search Using Time Series,” in *ECTI-CON 2021 - 2021 18th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology: Smart Electrical System and Technology, Proceedings*, Institute of Electrical and Electronics Engineers Inc., May 2021, pp. 171–174. doi: 10.1109/ECTI-CON51831.2021.9454824.

[10] C. Li, P. Ge, Z. Liu, and W. Zheng, “Forecasting Tourist Arrivals Using Denoising and Potential Factors,” *Ann Tour Res*, vol. 83, Jul. 2020, doi: 10.1016/j.annals.2020.102943.

[11] T. Lingyu, W. Jun, and Z. Chunyu, “Mode Decomposition Method Integrating Mode Reconstruction, Feature Extraction, and Elm for Tourist Arrival Forecasting,” *Chaos Solitons Fractals*, vol. 143, Feb. 2021, doi: 10.1016/j.chaos.2020.110423.

[12] E. Zhao, P. Du, and S. Sun, “Historical Pattern Recognition with Trajectory Similarity for Daily Tourist Arrivals Forecasting,” *Expert Syst Appl*, vol. 203, Oct. 2022, doi: 10.1016/j.eswa.2022.117427.

[13] S. Siami-Namini, N. Tavakoli, and A. Siami Namin, “A Comparison of Arima and LSTM in Forecasting Time Series,” in *Proceedings - 17th IEEE International Conference on Machine Learning and Applications, ICMLA 2018*, Institute of Electrical and Electronics Engineers Inc., Jul. 2018, pp. 1394–1401. doi: 10.1109/ICMLA.2018.00227.

[14] S. Junaidi et al., *Buku Ajar Machine Learning*. Jambi: Sonpedia, 2024.

[15] S. Rifky et al., *Artificial Intelligence (Teori Dan Penerapan AI di Berbagai Bidang)*. Jambi: PT. Sonpedia Publishing Indonesia, 2024.

[16] S. J. Russell and P. Norvig, *Artificial Intelligence A Modern Approach Third Edition*. New Jersey: Prantice Hall, 2010.

[17] Badan Pariwisata dan Ekonomi Kreatif, “Laporan Kinerja Kementerian Pariwisata dan Ekonomi Kreatif / Badan Pariwisata dan Ekonomi Kreatif,” Jakarta, 2023.

[18] Badan Pusat Statistik Provinsi NTB, “Provinsi Nusa Tenggara Barat Dalam Angka,” Mataram, 2023.

[19] A. A. Rizal and S. Soraya, “Multi Time Steps Prediction Dengan Recurrent Neural Network Long Short Term Memory,” *Matrik*, vol. 18, no. 1, pp. 115–124, 2018.

[20] A. A. Rizal and S. Hartati, “Recurrent Neural Network with Extended Kalman Filter for Prediction of The Number of Tourist Arrival in Lombok,” in *2016 International Conference on Informatics and Computing (ICIC)*, IEEE, 2016, pp. 180–185. doi: 10.1109/IAC.2016.7905712.