Published online on the journal's webpage: **http://journals.telkomuniversity.ac.id/index.php/dinda**

Journal of Dinda

Data Science, Information Technology, and Data Analytics

Vol. 6 No. 1 (2026) 25 - 34 E-ISSN: 2809-8064

Comparative Analysis of Deep Learning Models for Forecasting Visitors at Mount Rinjani

Zalikha Syahlarani Fathina Ahmad[1*], Putriaji Hendikawati[2]

[1*] Department of Mathematics, Faculty of Mathematics and Natural Sciences, Universitas Negeri Semarang

[2] Department of Mathematics, Faculty of Mathematics and Natural Sciences, Universitas Negeri Semarang

[1*]zalikhasyahlarani@students.unnes.ac.id, [2]putriaji.mat@mail.unnes.ac.id

Abstract

This study benchmarked five neural architectures (RNN, LSTM, FFNN, CNN, ANFIS) and two classical baselines (S-Naive, SARIMA) for forecasting daily visitor arrivals at Mount Rinjani National Park under operational constraints characteristic of regulated protected areas. Using 1,650 observations from January 2021 to July 2025, the dataset comprised 1,223 operational days averaging 150 visitors and 427 closures representing 25.9% of observations. Eight features were engineered following causal principles, including temporal indicators, lag variables, and rolling statistics. Hyperparameter optimization employed TimeSeriesSplit cross-validation with sMAPE as primary criterion, followed by 75:25 temporal seperate evaluation under dual scenarios, operational days only and combined operational-closure days. LSTM achieved superior performance with 9.73% operational MAPE, 11.45% combined sMAPE, MAE of 42.42 visitors, and RMSE of 72.51 visitors, substantially outperforming FFNN at 12.05% and 13.18%, RNN at 14.55% and 16.92%, CNN at 18.12% and 20.87%, S-Naive at 42.39% and 35.58%, SARIMA at 98.56% and 68.33%, and ANFIS at 33.78% and 38.92% for operational and combined scenarios respectively. Optimal configuration employed 64 units, 0.001 learning rate, 30-day lookback, and 0.2 dropout. SARIMA's catastrophic failure with 46.22 percentage point training-testing gap confirms classical methods' fundamental unsuitability for structural zeros. Findings establish LSTM's dual-task capability for continuous demand forecasting and operational state classification, enabling proactive visitor management supporting conservation objectives.

Keywords: *tourism forecasting, neural networks, time series, grid search*

1. Introduction

Tourism contributes 4.01 percent to GDP and employs 25.01 million workers in Indonesia as of 2024 [1]. Ecotourism has emerged as a strategic sub-sector that promotes conservation while enhancing local community welfare [2]. Mount Rinjani National Park (MRNP) recorded 189,091 visitors in 2024 [3], necessitating accurate demand forecasting for optimal management [4]. However, tourist arrival data exhibits non-linear, seasonal, and highly volatile patterns due to weather conditions and conservation policies. Classical baseline models such as Seasonal Naive and SARIMA have limitations in handling the complexity of these patterns [5], leading to uncertainty in operational planning.

Recent advances in neural and hybrid intelligence have demonstrated superior capability in capturing complex temporal patterns and non-linear relationships in tourism demand forecasting. Long Short-Term Memory (LSTM) networks address vanishing gradient problems through gating mechanisms, providing more stable predictions compared to traditional statistical models [6], [7], [8]. Recurrent Neural Networks (RNN) effectively capture long-term dependencies [9], while Feedforward Neural Networks (FFNN) achieve MAPE ranging from 7.9% to 9.2% across various tourism contexts [10], [11]. Convolutional Neural Networks (CNN) have demonstrated competitive or superior performance compared to traditional methods [12], and Adaptive Neuro-Fuzzy Inference Systems (ANFIS) prove effective for handling high-dimensional datasets [13], [14].

While recent advances demonstrate LSTM's effectiveness for tourism forecasting emerging architectures including transformer-based models and gradient boosting methods such as XGBoost have shown promise in other time series domains [23]. However,

Received: 29-01-2026 | Revised: 11-02-2026 | Published: 24-02-2026

these approaches typically require larger datasets and may not suit the dual-task forecasting challenge posed by ecotourism systems with structural operational constraints. This study focuses on established neural architectures suitable for limited-sample scenarios characteristic of national park contexts.

Robust hyperparameter optimization requires time series-specific validation strategies to prevent temporal leakage. TimeSeriesSplit cross-validation with expanding windows maintains chronological ordering where validation sets strictly follow training sets temporally, ensuring that future information never leaks into model selection [15]. Feature engineering must be performed causally, using only information available at prediction time such as calendar features, lag variables, and rolling statistics calculated from historical data [16], [17]. Despite these advances, comprehensive comparative studies evaluating multiple deep learning architectures simultaneously remain limited in tourism forecasting literature, particularly for national park ecotourism with seasonal restrictions, capacity constraints, and irregular closures that violate classical time series assumptions [18].

This research addresses these gaps through three key contributions. It provides systematic comparison of four neural networks and one hybrid intelligence system (RNN, LSTM, FFNN, CNN, and ANFIS) alongside two classical baselines (Seasonal Naive and SARIMA) for national park ecotourism forecasting with structural operational constraints, hyperparameter optimization for deep learning models employs TimeSeriesSplit cross-validation to prevent temporal leakage, while classical baselines follow standard statistical practices, Seasonal Naive requires no tuning, and SARIMA parameters are optimized using auto ARIMA with AIC minimization, also, it employs a dual evaluation framework that assesses performance separately on operational days only and combined operational-closure scenarios.

This study aims to forecast realized visitor arrivals at MRNP, focusing on actual daily visitor counts rather than potential demand. Closure periods are encoded as zero to reflect operational reality. Performance is assessed using RMSE, MAE, MAPE, and sMAPE [19], [20], with evaluation in two scenarios, operational days only to measure visitor volume prediction accuracy, and combined of both operational and non-operational days to assess capability in handling binary operational status alongside volume forecasting. Findings are expected to provide practical contributions for MRNP management and academic contributions for deep learning-based tourism forecasting in ecotourism contexts with structural operational constraints.

2. Research Methods

The research follows a structured workflow comprising data collection, feature engineering, temporal data splitting (75:25 train-test ratio), RobustScaler normalization, and grid search with TimeSeriesSplit cross-validation. The study evaluates five neural and hybrid intelligence models (RNN, CNN, FFNN, LSTM, and ANFIS) alongside two classical baselines (Seasonal Naive and SARIMA). Hyperparameter tuning uses MAPE and sMAPE as the primary criterion [15]. All features are calculated using only historical data to prevent temporal leakage. MAPE is calculated exclusively on operational days as it is undefined when actual values equal zero during closures, while sMAPE handles zero values symmetrically for combined-scenario evaluation. The complete methodology is illustrated in Figure 1.

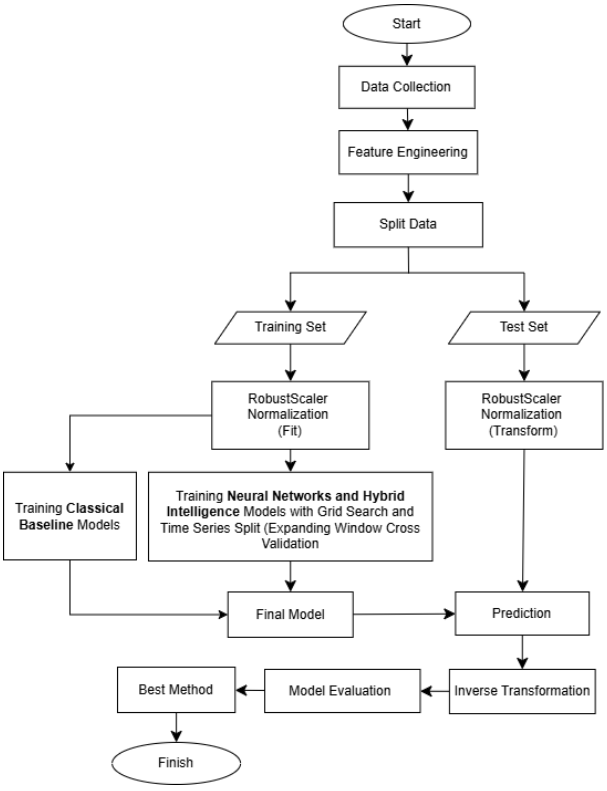

Figure 1. Research Procedures

2.1. *Data Source*

This study uses daily visitor data from Mount Rinjani National Park (MRNP) for the period January 1, 2021 to July 8, 2025, obtained from the official park website (https://www.rinjaninationalpark.id/grafik). The dataset contains 1,650 daily observations, comprising 1,223 operational days (when the park was open to visitors) and 427 closure days (including regular weekly closures, maintenance periods, and weather-related closures). MRNP is a conservation area covering 41,330 ha in

Lombok, West Nusa Tenggara, offering various tourist attractions such as mountain climbing, hot springs, and waterfalls.

2.2. *Variables and Feature Engineering*

The dependent variable in this study is the daily number of visitors (y_t) to MRNP. Following exploratory data analysis, feature engineering was performed to generate eight input features adhering to parsimony principles, where redundant or highly correlated features were eliminated to reduce dimensionality and prevent overfitting [16], [17]. Table 1 presents the constructed features.

Table 1. Constructed Features

No.	Feature	Type	Description
1	Operational	Binary	Operational status (1=open, 0=closed)
2	Day of The Week	Ordinal	Day of the week (Monday=0 - Sunday=6)
3	Weekend	Binary	Weekend indicator
4	High season	Binary	High season period (0=Jun - Aug)
5	Lag 1	Continuous	Number of visitors 1 day before
6	Lag 7	Continuous	Number of visitors 7 days before
7	Rolling Mean	Continuous	7 days moving average
8	Average	Continuous	Historical average per day

Features 1-4 are temporal features capturing calendar and seasonal patterns [4]. Features 5-7 are lag-based and rolling statistics functioning as autoregressive components [17], [21], while Feature 8 captures systematic day-of-week effects [22].

$$Lag_1(t) = y_{t-1}, \quad Lag_7(t) = y_{t-7} \tag{1}$$

Lag 1 (y_{t-1}) captures short-term persistence from the previous day, while Lag 7 (y_{t-7}) captures weekly seasonal patterns commonly observed in tourism demand [17].

$$Rolling\ Mean_7(t) = \frac{1}{7}\sum_{i=1}^{7} y_t - i \tag{2}$$

The 7-day rolling mean averages visitor counts from $t-7$ to $t-1$ (explicitly excluding day t) to smooth daily fluctuations while representing local trends [23].

$$Average_d(t) = \frac{1}{N_d}\sum_{\tau \epsilon Dd} y_\tau \tag{3}$$

The Average feature was calculated exclusively from the training set (January 2021 - April 2024) as the historical mean for each day of the week and held fixed during testing, preventing any test period information from influencing feature values. This ensures that when predicting May 2024 onwards, the model uses only historical day-of-week patterns available prior to the test period, maintaining strict temporal causality [22].

All features were computed causally using only information available prior to time t. Lag and rolling features used only past observations up to $t-1$. The Average feature was calculated exclusively from the training set as the historical mean for each day of the week and held fixed during testing, preventing any test period information from influencing feature values. During hyperparameter tuning, rolling features were recomputed within each TimeSeriesSplit fold to ensure out-of-sample validity [17].

2.3. *Data Preprocessing*

The preprocessing stage includes data cleaning, missing value handling, and normalization. Missing values arising from lag and rolling window calculations at the dataset start were removed because first 7 days lack complete historical context for 7-day features, resulting in 1,643 usable observations. Non-operational days due to ecotourism closures were coded as zero visitors to preserve the binary operational state structure critical for realistic forecasting evaluation. Normalization was performed using RobustScaler, which is resistant to outliers through quartile-based transformation:

$$y_{scaled} = \frac{y - Q1(y)}{Q3(y) - Q1(y)} \tag{4}$$

where $Q_1(y)$ and $Q_3(y)$ are the first and third quartiles of the dataset. was fitted exclusively on the training set and then apointlied to the test set to prevent data leakage. The data was split temporally with May 2024 as the cutoff point, resulting in 1,237 training samples (75%) and 413 test samples (25%). Temporal ordering was strictly preserved throughout all preprocessing and modeling stages to maintain chronological integrity [10].

2.4. *Recurrent Neural Network (RNN)*

RNN has a memory feature that allows output to be influenced by previous inputs through hidden states, making it suitable for sequential data such as time series. However, its main weakness is the problem of exponentially growing or decaying gradients in long sequences, which hinders the ability to capture long-range correlations [24]. The RNN architecture is defined in Formula (5).

$$h_t = \tanh(W_{hh}h_{t-1} + W_{xh}x_t + b_h) \tag{5}$$

where h_t is the hidden state at time t, x_t is the input, W are weight matrices, and b are bias vectors.

2.5. *Convolutional Neural Network (CNN)*

CNN extracts local patterns through convolution operations with architecture including input layer,

convolutional layer, pooling layer, flatten layer, and fully connected layer [24], [25]. This model excels in local pattern extraction with faster training compared to RNN or LSTM, but is less effective in capturing long-term dependencies [26]. The 1D convolution operation for time series can be expressed in Formula (6).

$$y_i = \sum_{k=0}^{K-1} W_k \cdot X_{i+k} + b \quad (6)$$

where w_k are kernel weights with size K, x_i is the input sequence, and b is the bias.

2.6. *Feed-Forward Neural Network (FFNN)*

FFNN is an Artificial Neural Network with unidirectional information flow trained through backpropagation and gradient descent. This model is stable for simple non-linear relationships but lacks memory, making it less effective for time series data with temporal dependencies [26]. The output layer computation as follows on Formula (7)

$$y_i = f(W_2 \cdot f(W_1 \cdot x + b_1) + b_2 \quad (7)$$

where f is the activation function (ReLU, sigmoid, or tanh), W_1, W_2 are weight matrices, and b_1, b_2 are bias vectors.

2.7. *Long-Short Term Memory (LSTM)*

LSTM is a development of RNN that overcomes the vanishing gradient problem through three gate mechanisms (forget, input, output) to regulate information flow [16]. This model can capture long-term dependencies and complex non-linear relationships in time series data [10]. The architecture consists of two LSTM layers, dropout layer, and dense layer.

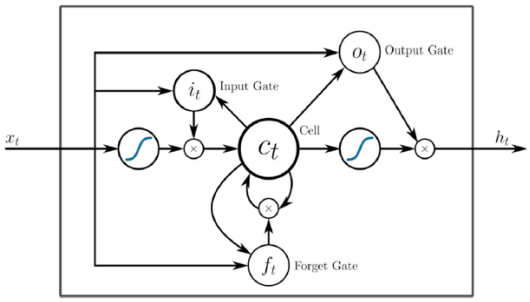

Figure 2. LSTM Structure

2.8. *Adaptive Neuro Fuzzy Inference System (ANFIS)*

ANFIS combines neural networks and fuzzy logic with Least Squares Estimator (LSE) and Error Backpropagation (EBP) learning. This model excels for complex and non-linear patterns with parameters: radius (0.5), squash factor (1.25), accept ratio (0.5), and reject ratio (0.15) [13], [14]. The ANFIS output is computed through a five-layer architecture with Formula (8)

$$y = \sum_i \overline{w}_i f_i = \sum_i \frac{w_i}{\sum_k w_k}(p_i x + q_i) \quad (8)$$

where w_i is the firing strength of the i-th rule, $\bar{w}_i$ is the normalized weight, and p_i, q_i are consequent parameters.

2.9. *Classical Baseline Models*

To ensure robust evaluation and contextualize the performance of neural and hybrid intelligence models, two classical time series baselines were implemented following best practices in forecasting evaluation [17], [22]. The first baseline is the Seasonal Naive (S-Naive) model, where the forecast is defined as:

$$\hat{y}_t = y_{t-7} \quad (9)$$

In this apointroach, the prediction for day t simply equals the actual value from the same day one week prior. This simple persistence model serves as a naive baseline that captures minimal weekly seasonality without parameter estimation. The second baseline is the SARIMA (Seasonal ARIMA) model, which follows the general form

$$\phi_P(B^S)\phi_P(B)\nabla_s^d\nabla^d Y_t = \theta_Q(B^S)\theta_q(B)\epsilon_t \quad (10)$$

where B is the backshift operator, $s = 7$ is the weekly seasonal period, and $\nabla_s^d\nabla^d$ are differencing operators. Model parameters $(p, d, q) \times (P, D, Q)_7$ were optimized viaauto ARIMA using AIC minimization with search bounds $p, q \leq 5$ and $P, Q \leq 2$ [22]. Unlike the neural and hybrid intelligence models that employ TimeSeriesSplit cross-validation for hyperparameter tuning, SARIMA parameters were selected via AIC minimization on the training set, representing a standard statistical modeling workflow. Together, these baselines ensure that neural network performance is evaluated against both simple and sophisticated statistical methods, following rigorous forecasting standards [17].

2.10. *Hyperparameter Tuning*

Hyperparameter optimization for neural and hybrid intelligence models employed grid search with TimeSeriesSplit cross-validation to prevent temporal leakage [15]. TimeSeriesSplit maintains chronological ordering through expanding windows where each validation set follows its training set temporally. The 1,237-sample training set was partitioned into three expanding folds, with each successive fold incorporating

more historical data for training while validating on the subsequent time period. Hyperparameter tuning was conducted on all days to optimize configurations for predicting realized visitor arrivals. The configuration achieving lowest validation sMAPE was selected as optimal. Table 2 presents the hyperparameter search spaces.

Table 2. Hyperparameter Configuration Space

Model	Hyperparameter	Search space
ANFIS	Membership Function	{2, 3}
	Learning rate	{0.001, 0.01}
	Batch size	{16, 32}
RNN, LSTM	Lookback window	{30, 45}
	Units	{32, 64, 128}
	Dropout rate	{0.2, 0.3}
	Learning rate	{0.0001, 0.001, 0.005}
	Epochs	{45, 100}
CNN	Lookback window	{30, 45}
	Units	{32, 64}
	Kernel size	{3, 5}
	Dropout rate	{0.2, 0.3}
	Epochs	{45, 100}
FFNN	Lookback window	{30, 45}
	Units	{32, 64, 128}
	Dropout rate	{0.2, 0.3}
	Learning rate	{0.001, 0.005}
	Epochs	{45, 100}

Hyperparameter search spaces were defined based on established practices in neural time series forecasting [15], [23]. Lookback windows (30, 45 days) capture monthly and extended seasonal patterns observed in the exploratory data analysis. Unit configurations balance model capacity with sample size (n=1,237 training samples) to prevent overfitting. Learning rates span typical ranges for Adam optimizer (0.0001-0.005) where values below 0.0001 cause slow convergence while values above 0.005 risk instability. Dropout rates (0.2-0.3) provide regularization without excessive information loss. Epoch limits (45-100) accommodate varying convergence speeds across architectures.

For classical baselines, Seasonal Naive requires no tuning as it employs a fixed rule-based prediction. SARIMA parameters were identified using auto ARIMA with AIC minimization on the training set, following standard statistical practice without cross-validation [22]. This distinction reflects different modeling paradigms: deep learning models require extensive hyperparameter search with out-of-sample validation, while classical statistical models rely on information criteria for parsimony. All optimized models were then evaluated on the held-out test set (413 days) using separate metrics for operational days and combined scenarios (operational + non-operational days).

2.11. *Model Evaluation*

Model performance was evaluated using multiple complementary metrics to provide a comprehensive assessment of forecasting accuracy: Root Mean Square Error (RMSE), Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE), and Symmetric Mean Absolute Percentage Error (sMAPE) [23], [26].

$$RMSE = \sqrt{\frac{1}{n}\sum_{t=1}^{n}(y_t - \hat{y}_t)^2} \quad (11)$$

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|x_i - x| \quad (12)$$

$$MAPE = \frac{\sum_{t=1}^{n}|\frac{y_t - \hat{y}_t}{y_t}|}{n} \times 100\% \quad (13)$$

$$sMAPE = \frac{1}{n}\sum_{t=1}^{n}\frac{|y_t - \hat{y}_t|}{(|y_t| + |\hat{y}_t|)/2} \times 100\% \quad (14)$$

where y_t is the actual value and $\hat{y}_t$ is the predicted value at time t, and n is the number of observations. RMSE penalizes large errors quadratically, making it sensitive to outliers, while MAE provides a linear error measure. MAPE expresses accuracy as an interpretable percentage with established thresholds: <10% (very good), 10-20% (good), 20-50% (fair), >50% (poor) [24].

MAPE becomes mathematically undefined when actual values equal zero due to division-by-zero. The MRNP dataset contains 427 closure days representing 25.9% of observations with zero visitors. Therefore, sMAPE was used as the primary criterion for hyperparameter selection because it remains defined even when actual values are zero [20]. MAPE was calculated on operational days only to provide interpretable percentage-based accuracy. For final evaluation, both MAPE (operational days) and sMAPE (all days) were reported to assess performance comprehensively. RMSE and MAE were calculated across all days as these absolute error metrics handle zero observations without mathematical issues.

3. Results and Discussion

Figure 3 illustrates MRNP visitor arrivals from January 2021 to July 2025, revealing high volatility and pronounced seasonality. Recurring peaks appear during June-August, coinciding with favorable weather and holidays periods that jointly create optimal outdoor recreation conditions [4]. The 2024-2025 period shows exceptional peaks exceeding 1,000 daily visitors versus moderate 2021-2022 volatility, suggesting post-pandemic demand recovery creating both economic opportunities and ecological challenges.

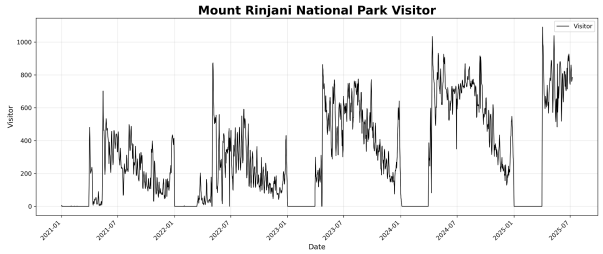

Figure 3. MRNP Visitor Time Series Plot

The dataset contains 1,650 observations, 1,223 represent operational days (150 visitors daily average) while 427 represent closures (25.9%). This structural zero proportion fundamentally distinguishes ecotourism from conventional tourism. National parks involve binary state transitions from conservation mandates, not continuous market dynamics. This creates dual challenges: predicting operational visitor volumes and identifying near-zero closure states. Classical methods assuming continuous stochastic processes are theoretically misaligned, motivating our dual evaluation framework assessing performance on operational days only and across all days [27].

3.1. *Hyperparameter Optimization Results*

Grid search explored 351 configurations across five models. Tables 3-7 present top performers by validation MAPE and sMAPE, revealing fundamental architectural strengths beyond simple accuracy.

Table 3. Grid Search Results of RNN Method

Units	Batch Size	LR	Epoch	MAPE	sMAPE
[128, 64, 32]	16	0.0001	100	14.69%	17.41%
[64, 32]	16	0.001	100	14.82%	17.57%
[128, 64, 32]	16	0.0005	100	21.14%	24.09%

Note: All best configurations have dropout rate of 0.2 and 30 Lookback

Table 4. Grid Search Results of CNN Method

Units	Batch Size	Kernel	Epoch	MAPE	sMAPE
[64, 32]	32	3	100	17.85%	21.23%
[32, 64]	32	5	100	21.12%	25.13%
[64, 32]	32	3	45	22.98%	27.36%

Note: All best configurations have dropout rate of 0.3 and 30 Lookback

Table 5. Grid Search Results of FFNN Method

Units	Batch Size	LR	Epoch	MAPE	sMAPE
[32, 64]	16	0.001	45	11.94%	14.08%
[64, 32]	16	0.001	100	12.23%	14.43%
[32, 64, 128]	32	0.005	45	21.16%	23.99%

Note: All best configurations have dropout rate of 0.2 and 30 Lookback

Table 6. Grid Search Results of LSTM Method

Units	Batch Size	LR	Epoch	MAPE	sMAPE
[64]	16	0.001	30	9.61%	11.30%
[64, 32]	16	0.001	45	10.28%	12.09%
[64]	32	0.0001	45	10.47%	12.31%

Note: All best configurations have dropout rate of 0.2 and 30 Lookback

Table 7. Grid Search Results of ANFIS Method

Membership Function	Batch size	LR	MAPE	sMAPE
3	16	0.001	33.08%	39.49%
2	16	0.001	33.46%	39.96%
3	32	0.01	35.93%	42.90%

Learning rate 0.001 proved critical but for different reasons. RNN showed extreme sensitivity, where a parameter value of 0.001 resulted in an error of 17.41%, while a smaller value of 0.0005 increased the error to 24.09%. due to gradient instability, multiplicative gradient flow creates exponential decay, so slower rates compound vanishing gradients across traversal steps. LSTM maintained robust consistency with not much difference in all architectures, because additive cell state updates preserve gradient magnitude, enabling rapid convergence with long-range learning as seen in [11]. FFNN exhibited similar learning rate sensitivity, where a smaller learning rate produced substantially better accuracy, while a larger learning rate led to a sharp decline in performance.

Dropout regularization of 0.2 proved optimal for LSTM, RNN, and FFNN, while CNN required higher regularization at 0.3 to prevent overfitting given its deeper convolutional structure. FFNN demonstrated remarkable stability across top configurations, with errors varying only 0.29-0.35 percentage points, between the two best performers, confirming robust generalization despite architectural simplicity. CNN exhibited the highest cross-validated errors among neural architectures with MAPE of 17.85% and sMAPE of 21.23%, reflecting architectural mismatch for temporal sequences where convolutional filters fail to capture long-range dependencies effectively [5].

ANFIS revealed fundamental limitations where varying membership functions from 2 to 3 provided negligible change in errors, while larger batch size of 32 degraded performance to 35.93% MAPE and 42.90% sMAPE. With 8 input features generating $2^8 = 256$ potential fuzzy rules but only 1,237 training samples, the resulting rule sparsity of apointroximately 4.8 samples per rule prevents effective membership function optimization, consistent with curse of dimensionality effects documented in fuzzy systems literature [14]. Cross-validated results confirm ANFIS is unsuited for this high-dimensional tourism forecasting task.

3.2. *Model Performance Comparison*

Following hyperparameter optimization via TimeSeriesSplit cross-validation, optimal configurations were evaluated on the held-out test set. Table 8 assesses regression performance on operational days using MAPE and sMAPE, while Table 9 presents the primary evaluation on combined scenarios using sMAPE.

Table 8. Model Performance Comparison on Operational Days

Model	RMSE	MAE	sMAPE	MAPE Training	MAPE Testing
S-Naïve	208.56	138.89	29.84%	-	42.39%
SARIMA	630.68	586.73	68.33%	52.34%	98.56%
RNN	106.20	75.48	16.92%	12.34%	14.55%
CNN	132.98	93.25	20.87%	15.89%	18.12%
FFNN	77.27	54.65	13.18%	11.18%	12.05%
LSTM	**72.51**	**42.42**	**11.45%**	**8.12%**	**9.73%**
ANFIS	246.99	212.98	38.92%	34.23%	33.78%

Note: Bold values indicate best performance per metric.

LSTM achieved superior performance with a modest generalization gap, demonstrating robust learning through gating mechanisms that suppress noise while retaining important temporal patterns, which naturally prevents overfitting through selective memory updating. [6], [13]. FFNN ranked second by learning generalized nonlinear patterns from engineered lag features, effectively transforming forecasting into static pattern recognition without the overfitting risks of sequential models. RNN and CNN showed moderate overfitting due to gradient instability and architectural mismatch with temporal dependencies, limiting their ability to learn long-range seasonal patterns [5]. ANFIS primarily demonstrated severe underfitting, where the model failed to capture even training patterns, Poorly-calibrated rules generate predictions uncorrelated with actual patterns, with slight testing "improvement" reflecting random variation rather than learning.

SARIMA showed extreme overfitting, where its continuous assumptions failed in the presence of structural zeros, making it unable to handle closure-like conditions in the data [27]. This is not overfitting in the traditional sense but rather architectural unsuitability for dual-state forecasting.

Table 9. Model Performance Comparison on All Days

Model	RMSE	MAE	sMAPE
S-Naïve	180.22	102.53	35.58%
SARIMA	630.68	586.73	68.33%
RNN	106.20	75.48	16.92%
CNN	132.98	93.25	20.87%
FFNN	77.27	54.65	13.18%
LSTM	**72.51**	**42.42**	**11.45%**
ANFIS	246.99	212.98	38.92%

Note: All days evaluation includes both operational days (n=350) and non-operational days (n=91)

LSTM achieved superior performance and modest generalization gap of 1.61 points, demonstrating robust learning [13]. forget, input, and output gates act as learned filters suppressing noise while retaining salient patterns. This architectural bias toward parsimonious representations prevents overfitting through selective memory updating, as demonstrated by Salamanis et al. [6]. FFNN ranked second by learning nonlinear patterns from lag features, supporting static pattern recognition [11]. RNN and CNN showed moderate overfitting with gaps exceeding 2.2 points due to vanishing gradients and architectural mismatch [5], [20].

ANFIS showed relatively high error and a slight negative performance gap, highlighting limitations from the curse of dimensionality where 256 fuzzy rules but only 1,237 training samples prevent effective learning [14].

The baselines demonstrated contrasting performance. S-Naive employed 7-day lag forecasting as a simple weekly persistence baseline, while $SARIMA(2,1,2)(1,1,2)_7$ exhibited severe failure with a large performance gap. This degradation occurs because SARIMA's continuous Gaussian assumptions are fundamentally violated by frequent structural zeros, causing the model to learn normal operational patterns during training but systematically predict positive visitor counts when encountering closure conditions in testing data [27].

All days evaluation reflects the real forecasting challenge for park managers because it includes structural zeros from conservation policies, unlike operational-day results that only show continuous demand behavior. LSTM's low error across both operational and closure conditions demonstrates practical precision for predictive management. The large performance gap between neural networks and classical methods confirms that ecotourism forecasting under operational constraints requires models capable of learning both continuous demand patterns and discrete operational states.

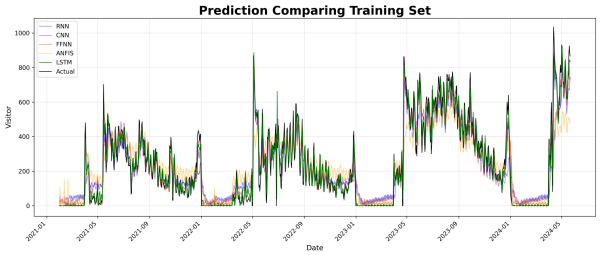

Figure 4. Prediction Comparison Training Set

Figure 4 highlights distinct learning behaviors across models. LSTM closely follows daily and seasonal variations while correctly predicting near-zero values during closures, reflecting effective memory control through its gating mechanisms. FFNN shows a strong fit with slight smoothing from global pattern learning. RNN

displays lagged predictions due to difficulty capturing long-term dependencies, while CNN exhibits oscillations from limited receptive fields. ANFIS consistently underestimates demand, indicating persistent underfitting across varying visitor ranges.

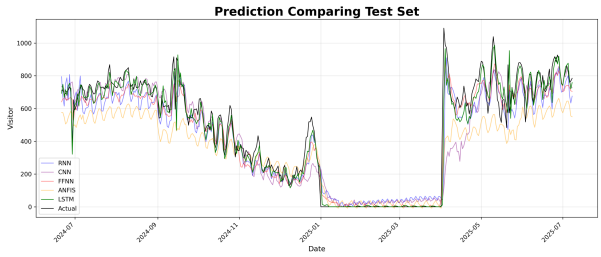

Figure 5. Prediction Comparing Test Set

Figure 5 confirms strong generalization on unseen data, where LSTM maintains superior performance despite novel patterns, indicating that its gating mechanisms learned genuine causal relationships. Its continued ability to predict near-zero values during test closures further demonstrates learned conditional logic that suppresses demand forecasts when operational signals indicate closures.

LSTM stands out because selective memory via gating mechanisms retains long-term seasonal dependencies while responding to short-term fluctuations. Vanishing gradient mitigation through additive cell state updates shown in Eq. (12) prevents gradient decay, explaining superiority over RNN [9].

$$(C_t = f_t \odot C_{\{t-1\}} + i_t \odot \tilde{C}_t) \qquad (15)$$

Adaptive feature weighting allows gates to prioritize lag variables for weekdays and high season indicators for peaks.

Comparative evaluation requires acknowledging our 11.45% sMAPE on combined data addresses a forecasting challenge absent in existing tourism literature assuming continuous demand. This study's LSTM achieved 9.73% operational MAPE, substantially outperforming Nguyen-Da et al. [27]'s hybrid CNN-LSTM at 12.3% and aligning with systematic reviews showing LSTM typically achieves 0.71-10% MAPE. Classical methods reveal fundamental limitations, while Bouhaddour et al. [7] reported SARIMA achieving 15-25% higher MAPE than LSTM, SARIMA exhibited model failure because continuous Gaussian assumptions cannot accommodate binary operational states. CNN's 18.12% and ANFIS's 33.78% operational MAPE confirm Kim et al. [5] and Karaboga and Kaya [14]'s reported architectural limitations.

The research establishes comprehensive benchmark comparing five neural and hybrid intelligence models for national park ecotourism forecasting. LSTM enables proactive management through accurate forecasts, supporting staffing optimization and carrying capacity interventions when predicted surges exceed 850 daily visitors. Benefits include reduced operational inefficiencies, optimized procurement, ecosystem protection through predictive conservation, and improved visitor experience through public dissemination of crowding forecasts for informed trip planning.

4. Conclusion

This study benchmarked five neural and hybrid intelligence models with two classical baselines for forecasting Mount Rinjani National Park visitor arrivals under operational constraints. Using 1,650 observations with TimeSeriesSplit cross-validation, we evaluated performance on operational days and combined scenarios including 427 closures (25.9%). LSTM showed superior performance with 9.73% operational MAPE and 11.45% combined sMAPE, alongside a MAE of 42.42 visitors and RMSE of 72.51 visitors, supporting practical resource allocation. This performance was achieved using 64 units, a 0.001 learning rate, a 30-day lookback window, and 0.2 dropout, where gating mechanisms enable selective memory use for both continuous demand forecasting and operational state recognition.

These results highlight the superiority of deep learning, particularly LSTM over other modeling approaches in capturing both continuous demand patterns and discrete operational states for conservation-oriented forecasting. Future research could integrate exogenous variables through multivariate LSTM architectures, apply ensemble approaches for uncertainty quantification, and extend validation across multiple protected areas to build transferable predictive management frameworks that support conservation objectives. Expanding temporal evaluation would also help assess adaptability to evolving visitation patterns. Incorporating additional predictors may further enhance performance. Broader multi-site testing could improve generalizability beyond location-specific operational dynamics. While feature parsimony reduced overfitting risk given limited training data, richer interactions remain worth exploring, alongside more extensive testing designs to strengthen statistical comparison and overall robustness.

Acknowledgments

The authors would like to express sincere gratitude to Mount Rinjani National Park (MRNP) authorities for providing access to visitor data essential for this research. Special apointreciation is given to the editor and anonymous reviewers for their insightful comments and recommendations, which have substantially improved this work.

References

[1] Office of Assistant to Deputy Cabinet Secretary for State Documents & Translation, "Indonesia Tourism Sector Shows Positive Growth in 2024," 2025. [Online]. Available: https://setkab.go.id/en/indonesia-tourism-sector-shows-positive-growth-in-2024/. Accessed: Nov. 05, 2025.

[2] M. Novia and Katno, "Ecotourism development as a sustainable tourism development strategy in Central Java," in Proc. 3rd Multidiscip. Int. Conf., Jakarta, Indonesia, Oct. 2023. doi: 10.4108/eai.28-10-2023.2341746.

[3] M. Nasir, "189.091 wisatawan kunjungi Gunung Rinjani, negara cuan Rp22,5 miliar," IDN Times NTB, 2025. [Online]. Available: https://ntb.idntimes.com/news/ntb/muhammad-nasir-18/189-091-wisatawan-kunjungi-gunung-rinjani-negara-cuan-rp22-5-miliar-00-ldn3d-vslnvd. Accessed: Nov. 01, 2025.

[4] A. F. Fithriyah and N. Shimizu, "Forecasting demand factors of tourist arrivals in Indonesia's tourism industry using recurrent neural network," IOP Conf. Ser.: Mater. Sci. Eng., vol. 1077, no. 1, art. no. 012035, 2021. doi: 10.1088/1757-899X/1077/1/012035.

[5] D. Kim, D. Kim, and K. Kim, "A daily tourism demand prediction framework based on multi-head attention CNN: The case of the foreign entrant in South Korea," Expert Syst. Appl., vol. 192, art. no. 116344, 2022. doi: 10.1016/j.eswa.2021.116344.

[6] A. Salamanis, G. Xanthopoulou, D. Kehagias, and D. Tzovaras, "LSTM-based deep learning models for long-term tourism demand forecasting," Electronics, vol. 11, no. 22, art. no. 3681, Nov. 2022. doi: 10.3390/electronics11223681.

[7] S. Bouhaddour, M. Sbihi, F. Guerouate, and C. Saadi, "Assessing tourism prediction models: A comparative study of SARIMA, random forest, and LSTM, considering nonlinear trends and the influence of COVID-19," Ingénierie des Systèmes d'Information, vol. 29, no. 6, pp. 2443–2454, 2024. doi: 10.18280/isi.290625.

[8] M. Istaltofa, Sarwido, and A. Sucipto, "Comparison of linear regression and LSTM (long short-term memory) in cryptocurrency prediction," J. Dinda: Data Sci., Inf. Technol., Data Anal., vol. 4, no. 2, pp. 141–148, Aug. 2024. doi: 10.20895/dinda.v4i2.1575.

[9] J. Lemmel, Z. Babaiee, M. Kleinlehner, I. Majic, P. Neubauer, J. Scholz, R. Grosu, and S. A. Neubauer, "Deep-learning vs regression: Prediction of tourism flow with limited data," arXiv:2206.13274, Jun. 2022. [Online]. Available: https://arxiv.org/abs/2206.13274. Accessed: Nov. 07, 2025.

[10] L. Q. Nguyen, P. O. Fernandes, and J. P. Teixeira, "Analyzing and forecasting tourism demand in Vietnam with artificial neural networks," Forecasting, vol. 4, no. 1, pp. 36–50, Jan. 2022. doi: 10.3390/forecast4010003.

[11] A. Alamsyah, P. Bella, and A. Friscintia, "Artificial neural network for Indonesian tourism demand forecasting," in Proc. 7th Int. Conf. Inf. Commun. Technol. (ICoICT), Kuala Lumpur, Malaysia, Jul. 2019, pp. 1–7. doi: 10.1109/ICoICT.2019.8835382.

[12] K. Kartik, T. Ahmed, S. Ghosh, and R. Gupta, "Tourist prediction for Chicago using CNN (convolutional neural network)," Int. J. Trend Sci. Res. Dev., vol. 7, no. 5, pp. 1100–1108, Sep.–Oct. 2023. doi: 10.5281/zenodo.15515494.

[13] S. Cankurt and A. Subasi, "Tourism demand forecasting using stacking ensemble model with adaptive fuzzy combiner," Soft Comput., vol. 26, no. 7, pp. 3455–3467, Apr. 2022. doi: 10.1007/s00500-021-06695-0.

[14] D. Karaboga and E. Kaya, "Estimation of number of foreign visitors with ANFIS by using ABC algorithm," Soft Comput., vol. 24, no. 10, pp. 7579–7591, May 2020. doi: 10.1007/s00500-019-04386-5.

[15] V. Cerqueira, L. Torgo, and I. Mozetič, "Evaluating time series forecasting models: An empirical study on performance estimation methods," Mach. Learn., vol. 109, no. 11, pp. 1997–2028, Nov. 2020. doi: 10.1007/s10994-020-05910-7.

[16] K. He, L. Ji, C. W. D. Wu, and K. F. Tso, "Using SARIMA–CNN–LSTM approach to forecast daily tourism demand," J. Hosp. Tour. Manag., vol. 49, pp. 25–33, Dec. 2021. doi: 10.1016/j.jhtm.2021.08.022.

[17] S. R. Malla, S. Kayastha, R. Suwal, H. C. Bhandari, and R. Adhikari, "XGBoost forecasting of NEPSE index log returns with walk-forward validation," arXiv:2601.08896, Jan. 2026. [Online]. Available: https://arxiv.org/abs/2601.08896. Accessed: Feb. 07, 2026.

[18] B. Guan, E. S. Silva, H. Hassani, and S. Heravi, "Forecasting tourism growth with state-dependent models," Ann. Tour. Res., vol. 94, art. no. 103385, May 2022. doi: 10.1016/j.annals.2022.103385.

[19] C. Tofallis, "A better measure of relative prediction accuracy for model selection and model estimation," J. Oper. Res. Soc., vol. 66, no. 8, pp. 1352–1362, Aug. 2015. doi: 10.1057/jors.2014.103.

[20] D. Chicco, M. J. Warrens, and G. Jurman, "The coefficient of determination R-squared is more informative than SMAPE, MAE, MAPE, MSE and RMSE in regression analysis evaluation," PeerJ Comput. Sci., vol. 7, art. no. e623, Jul. 2021. doi: 10.7717/peerj-cs.623.

[21] R. Law, G. Li, D. K. C. Fong, and X. Han, "Tourism demand forecasting: A deep learning approach," Ann. Tour. Res., vol. 75, pp. 410–423, Mar. 2019. doi: 10.1016/j.annals.2019.01.014.

[22] A. Ampountolas, "Modeling and forecasting daily hotel demand: A comparison based on SARIMAX, neural networks, and GARCH models," Forecasting, vol. 3, no. 3, pp. 580–595, Aug. 2021. doi: 10.3390/forecast3030037.

[23] B. Lim and S. Zohren, "Time-series forecasting with deep learning: A survey," Phil. Trans. R. Soc. A, vol. 379, no. 2194, art. no. 20200209, Mar. 2021. doi: 10.1098/rsta.2020.0209.

[24] N. N. Lingga, Indwiarti, and A. A. Rohmawati, "Pemodelan dan peramalan kedatangan wisatawan ke tempat wisata dengan Google Trends menggunakan metode variasi

kalender ARIMAX," e-Proceeding of Engineering, vol. 8, no. 2, pp. 3361–3368, Apr. 2021. [Online]. Available: https://openlibrary.telkomuniversity.ac.id.

[25] A. Agga, A. Abbou, M. Labbadi, and Y. El Houm, "Short-term self consumption PV plant power production forecasts based on hybrid CNN-LSTM, ConvLSTM models," Renew. Energy, vol. 177, pp. 101–112, Nov. 2021. doi: 10.1016/j.renene.2021.05.095.

[26] M. A. Syarifudin, D. C. R. Novitasari, F. Marpaung, N. Wahyudi, D. P. Hapsari, E. Supriyati, Y. Farida, F. M. Amin, R. R. D. Nugraheni, and R. Nariswari, "Hotspot prediction using 1D convolutional neural network," Procedia Comput. Sci., vol. 179, pp. 845–853, 2021. doi: 10.1016/j.procs.2021.01.073.

[27] T. Nguyen-Da, Y.-M. Li, C.-L. Peng, M.-Y. Cho, and P. Nguyen-Thanh, "Tourism demand prediction after COVID-19 with deep learning hybrid CNN–LSTM—case study of Vietnam and provinces," Sustainability, vol. 15, no. 9, art. no. 7179, Apr. 2023. doi: 10.3390/su15097179.